



**TUGAS AKHIR - KI141502**

# **SISTEM MONITORING VOLUME CAIRAN DI DALAM TANGKI BERBASIS INTEGRASI SENSOR ULTRASONIK PADA MIKROKONTROLER ARDUINO**

**YOGA PRATAMA ALIARHAM**  
**NRP 5111100018**

**Dosen Pembimbing I**  
**Ir. Muchammad Husni, M.Kom**

**Dosen Pembimbing II**  
**Henning Titi C, S.Kom, M.Kom**

**JURUSAN TEKNIK INFORMATIKA**  
**Fakultas Teknologi Informasi**  
**Institut Teknologi Sepuluh Nopember**  
**Surabaya 2015**



**UNDERGRADUATE THESES - KI141502**

# **LIQUID TANK'S VOLUME MONITORING SYSTEM BASED ON INTEGRATED ULTRASONICS SENSOR AT ARDUINO MICROCONTROLLER**

**YOGA PRATAMA ALIARHAM  
NRP 5111100018**

**Supervisor I  
Ir. Muchammad Husni, M.Kom**

**Supervisor II  
Henning Titi C, S.Kom, M.Kom**

**DEPARTMENT OF INFORMATICS  
FACULTY OF INFORMATION TECHNOLOGY  
INSTITUT TEKNOLOGI SEPULUH NOPEMBER  
SURABAYA 2015**

## LEMBAR PENGESAHAN

### SISTEM MONITORING VOLUME CAIRAN DI DALAM TANGKI BERBASIS INTEGRASI SENSOR ULTRASONIK PADA MIKROKONTROLER ARDUINO

#### TUGAS AKHIR

Diajukan Untuk Memenuhi Salah Satu Syarat  
Memperoleh Gelar Sarjana Komputer  
pada  
Bidang Studi Komputasi Berbasis Jaringan  
Program Studi S-1 Jurusan Teknik Informatika  
Fakultas Teknologi Informasi  
Institut Teknologi Sepuluh Nopember

Oleh

**YOGA PRATAMA ALIARHAM**

**NRP : 5111 100 018**

Disetujui oleh Dosen Pembimbing Tugas Akhir

1. Ir. Muchammad Husni, M.Kom  
NIP: 196002211984031001

(Pembimbing 1)

2. Henning Titi C, S.Kom, M.Kom  
NIP: 198407082010122004

(Pembimbing 2)

**SURABAYA  
JUNI, 2015**

# **SISTEM MONITORING VOLUME CAIRAN DI DALAM TANGKI BERBASIS INTEGRASI SENSOR ULTRASONIK PADA MIKROKONTROLER ARDUINO**

**Nama Mahasiswa : YOGA PRATAMA ALIARHAM**  
**NRP : 5111100018**  
**Jurusan : Teknik Informatika FTIF-ITS**  
**Dosen Pembimbing 1 : Ir. Muchammad Husni, M.Kom**  
**Dosen Pembimbing 2 : Henning Titi C, S.Kom, M.Kom**

## ***Abstrak***

*Kasus pencurian minyak dengan cara mengambil sebagian muatan dari kapal tanker di Indonesia masih menjadi berita terkini. Salah satu contohnya adalah kasus pencurian minyak di Kepulauan Riau. Pada kasus tersebut, pelaku berhasil mendapatkan uang sebesar 1,3 triliun. Tindakan pencurian muatan kapal tanker ini dilakukan dengan cara “mengencingkan” muatan kapal tanker ke kapal illegal, sehingga perlu melibatkan kru kapal tanker untuk melakukan perbuatan tersebut.*

*Untuk mencegah perbuatan kriminal tersebut supaya tidak terjadi terus menerus, maka dibutuhkan mekanisme untuk memonitoring kendaraan bermuatan BBM dari jauh, sehingga pihak yang berada dikendaraan tersebut tidak dapat bekerjasama dengan pencuri muatan.*

*Tugas akhir ini memanfaatkan mikrokontroler arduino, modul GPS/GPRS/GSM Shield, modul bluetooth dan sensor ultrasonik untuk membuat sebuah jaringan sensor nirkabel sederhana, yang dapat digunakan untuk mengkalkulasikan data ketinggian cairan dan mendapatkan data lokasi kendaraan bermuatan BBM, sehingga kendaraan tersebut dapat dimonitoring dari jauh.*

*Dari uji coba komponen sensor, didapatkan tingkat akurasi terbaik sebesar 99.33% jika kendaraan berada pada medan yang normal, dan didapatkan akurasi sebesar 84 % jika kendaraan berada pada tanjakan atau turunan, sehingga komponen ini dapat digunakan untuk memonitoring kendaraan bermuatan BBM dari pos SPBU atau dari pelabuhan, untuk mencegah terjadinya pencurian illegal terhadap kendaraan tersebut.*

***Kata kunci: Monitoring Tangki, Arduino, Sensor Ultrasonik, GPS/GPRS/GSM Shield V3.0***

# **LIQUID TANK'S VOLUME MONITORING SYSTEM BASED ON INTEGRATED ULTRASONICS SENSOR AT ARDUINO MICROCONTROLLER**

**Student's Name : YOGA PRATAMA ALIARHAM**  
**Student's ID : 5111100018**  
**Department : Teknik Informatika FTIF-ITS**  
**First Advisor : Ir. Muchammad Husni, M.Kom**  
**Second Advisor : Henning Titi C, S.Kom, M.Kom**

## ***Abstract***

*In Indonesia, the criminal activity about stealing a part of tanker's freight is still the latest news. For example, in the Riau case, the culprit were success to steal about 1.3 billion rupias from stealing just a part of the freight. They do so by "urinate" the freight little by little to another ship, so that they have to work together with the tanker's crew.*

*To prevent that criminal activity, it is necessary to design a mechanism so that the vehicle can be monitored from afar, so the vehicle's crew can not do joint operation with the culprit.*

*This undergraduated thesis uses arduino microcontroller, GPS/GPRS/GSM module, bluetooth module, and ultrasonic sensor to create a simple wireless sensor network that can be used to calculate the liquid level of tha tank and exact position of the vehicle so the vehicle can be monitored from afar. If the vehicle can be monitored from afar, so we can prevent the criminal activity occur on that vehicle.*

*From few trials above, this component obtain 99.33 % accuracy rate if the vehicle is on the normal track, and obtain 84 % accuracy rate if the vehicle is on the ramp track, so that this component can be used to monitor the gasoline tank from the*

*SPBU station or form the port, to prevent theft charge on that vehicle.*

***Keyword: Tank Monitoring, Arduino, Ultrasonic Sensor, GPS/GPRS/GSM Shield V3.0***

## KATA PENGANTAR

بِسْمِ اللَّهِ الرَّحْمَنِ الرَّحِيمِ

Alhamdulillahirabbil'alamin, segala puji bagi Allah SWT, yang telah melimpahkan rahmat dan hidayah-Nya sehingga penulis dapat menyelesaikan Tugas Akhir yang berjudul **“SISTEM MONITORING VOLUME CAIRAN DI DALAM TANGKI BERBASIS INTEGRASI SENSOR ULTRASONIK PADA MIKROKONTROLER ARDUINO”**. Shalawat serta salam selalu senantiasa saya tujukan kepada Rasulullah SAW, selaku inspirator dunia nomor satu saat ini, hingga akhir zaman.

Pengerjaan Tugas Akhir ini merupakan salah satu dari sekian banyak kesempatan yang saya dapatkan, untuk mendapatkan ilmu dan pengalaman berharga selama saya berada di kampus Teknik Informatika ITS ini. Dengan pengerjaan Tugas Akhir ini, saya menjadi semakin bisa untuk memanajemen waktu dan memanajemen diri sendiri, sehingga Tugas Akhir ini dapat selesai tepat waktu.

Selesainya Tugas Akhir ini tidak lepas dari bantuan dan dukungan beberapa pihak. Sehingga pada kesempatan ini penulis mengucapkan syukur dan terima kasih kepada:

1. Allah SWT dan Nabi Muhammad SAW.
2. Ibu, Ibu, Ibu, Ayah dan Adik yang selalu memberikan do'a, dukungan, serta motivasi, sehingga penulis selalu termotivasi untuk menyelesaikan Tugas Akhir.
3. Bapak Ir. Muchammad Husni, M.Kom, selaku pembimbing I yang selalu menyemangati dan memotivasi dengan ilmu-ilmu yang diluar dugaan saya.
4. Ibu Henning Titi C, S.Kom, M.Kom selaku pembimbing II yang telah mengoreksi buku ini dengan cermat.
5. Ibu Dr. Eng. Nanik Suciati, S.Kom., M.Kom. selaku Kepala Jurusan Teknik Informatika ITS, Bapak Radityo Anggoro, S.Kom.,M.Sc. selaku koordinator TA yang selalu meluangkan



waktunya untuk *sharing* ilmu bersama saya, dan segenap dosen Teknik Informatika yang telah memberikan ilmunya.

6. Teman – Teman Kabinet Bersahabat HMTTC yang telah mengisi waktu-waktu penulis penuh dengan inspirasi dan kenangan yang tidak akan bisa terlupakan.
7. Teman – teman TC Hura Hura, yang telah menemani penulis untuk berpetualang bersama mengarungi pulau Jawa, sampai saat ini.
8. Teman – teman angkatan 2011, tanpa mereka, saya tidak akan merasakan apa itu yang dinamakan “Angkatan”.
9. Teman – teman administrator Laboratorium Manajemen Informasi, yang telah memberikan saya kesempatan untuk mempelajari lebih dalam tentang MI, serta teman lab lain yang telah mengajari saya banyak sekali, iya, banyak sekali hal secara tidak langsung.
10. Serta semua pihak yang telah turut membantu penulis dalam menyelesaikan Tugas Akhir ini

Penulis menyadari bahwa Tugas Akhir ini masih memiliki banyak kekurangan. Sehingga dengan kerendahan hati, penulis mengharapkan kritik dan saran dari pembaca untuk perbaikan ke depannya.

Surabaya, Juni 2015

## DAFTAR ISI

LEMBAR PENGESAHAN .....	v
<i>Abstrak</i> .....	vii
<i>Abstract</i> .....	ix
KATA PENGANTAR .....	xi
DAFTAR ISI .....	xiii
DAFTAR GAMBAR .....	xv
DAFTAR TABEL .....	xxi
BAB I PENDAHULUAN .....	1
1.1 Latar Belakang .....	1
1.2 Rumusan Masalah .....	2
1.3 Batasan Masalah .....	2
1.4 Tujuan .....	3
1.5 Manfaat .....	3
1.6 Metodologi .....	3
1.7 Sistematika Penulisan Laporan Tugas Akhir .....	4
BAB II TINJAUAN PUSTAKA .....	7
2.1 Mikrokontroler Arduino .....	7
2.2 Sensor Ultrasonik .....	11
2.3 GPS ( <i>Global Positioning System</i> ) .....	12
2.4 GPS/GSM/GPRS Shield Versi 3.0 .....	14
2.5 Modul <i>Bluetooth</i> HC 05 & HC 06 .....	15
2.6 <i>Leaflet.js</i> API .....	16
BAB III DESAIN PERANGKAT .....	17
3.1 Perancangan Integrasi Mikrokontroler dan Komponen Sensor .....	17
3.1.1 Perancangan <i>Child Node</i> .....	18
3.1.2 Perancangan <i>Center Node</i> .....	20
3.2 Perancangan Aplikasi Monitoring .....	22
3.2.1 Perancangan Diagram <i>Use Case</i> .....	23
3.2.2 Perancangan Basis Data .....	24
3.2.3 Perancangan <i>Web Service</i> .....	26

3.2.4	Fungsi <i>getVehicleStatus</i> .....	26
3.2.5	Perancangan Antarmuka Sistem .....	27
BAB IV IMPLEMENTASI .....		31
4.1	Lingkungan Implementasi .....	31
4.2	Implementasi .....	31
4.2.1	Implementasi Mikrokontroler dan Komponen Sensor .....	32
4.2.2	Implementasi Aplikasi Monitoring .....	38
BAB V UJI COBA DAN EVALUASI .....		45
5.1	Lingkungan Uji Coba .....	45
5.2	Skenario Uji Coba .....	46
5.3	Analisa Hasil Uji Coba .....	63
BAB VI KESIMPULAN DAN SARAN .....		65
6.1	Kesimpulan .....	65
6.2	Saran .....	65
DAFTAR PUSTAKA .....		67
LAMPIRAN .....		69
BIODATA PENULIS .....		71

## DAFTAR TABEL

<i>Tabel 2.1. AT Command pada GPS/GPRS/GSM Shield V3.0 .....</i>	<i>14</i>
<i>Tabel 2.2. AT Command pada Bluetooth HC 05 dan HC 06 .....</i>	<i>15</i>
<i>Tabel 3.1. Perancangan tabel vehicle_master .....</i>	<i>24</i>
<i>Tabel 3.2. Perancangan tabel user.....</i>	<i>25</i>
<i>Tabel 3.3. Perancangan tabel initial_data .....</i>	<i>25</i>
<i>Tabel 3.4. Perancangan tabel track_record.....</i>	<i>26</i>
<i>Tabel 5.1. Data ketinggian yang berhasil dikirim ke server pada uji coba 1 .....</i>	<i>55</i>
<i>Tabel 5.2. Data lokasi yang berhasil dikirim ke server pada uji coba 1 .....</i>	<i>56</i>
<i>Tabel 5.3. Data ketinggian yang berhasil dikirim ke server pada uji coba 2 .....</i>	<i>58</i>
<i>Tabel 5.4. Data lokasi yang berhasil dikirim ke server pada uji coba 1 .....</i>	<i>59</i>
<i>Tabel 5.5. . Data yang berhasil dikirim ke server pada uji coba 3 .....</i>	<i>61</i>
<i>Tabel 5.6. Perbandingan tingkat akurasi percobaan 1 dan 2 .....</i>	<i>63</i>
<i>Tabel A.1. Hasil Analisa Uji Coba Skenario 1 .....</i>	<i>69</i>
<i>Tabel A.2. Hasil Analisa Uji Coba Skenario 2 .....</i>	<i>70</i>

## DAFTAR GAMBAR

<i>Gambar 2.1. Mikrokontroler Arduino .....</i>	<i>7</i>
<i>Gambar 2.2. Arduino UNO .....</i>	<i>8</i>
<i>Gambar 2.3. Arduino Due .....</i>	<i>9</i>
<i>Gambar 2.4. Arduino Leonardo .....</i>	<i>9</i>
<i>Gambar 2.5. Arduino Fio .....</i>	<i>10</i>
<i>Gambar 2.6. Arduino Mega.....</i>	<i>10</i>
<i>Gambar 2.7. Sensor Ultrasonik.....</i>	<i>12</i>
<i>Gambar 2.8. Butuh minimal 3 satelit untuk mendapatkan koordinat fix dari GPS.....</i>	<i>13</i>
<i>Gambar 2.9. Penggambaran peta dunia dengan menggunakan API leaflet.js .....</i>	<i>16</i>
<i>Gambar 3.1. Rancangan integrasi alat secara umum .....</i>	<i>18</i>
<i>Gambar 3.2. Perancangan Child node.....</i>	<i>19</i>
<i>Gambar 3.3. Protokol pengiriman dari child node .....</i>	<i>20</i>
<i>Gambar 3.4. Perancangan Center Node .....</i>	<i>20</i>
<i>Gambar 3.5. Data mentah yang diterima oleh modul GPS/GSM/GPRS Shield.....</i>	<i>21</i>
<i>Gambar 3.6. Alur proses dari center node .....</i>	<i>22</i>
<i>Gambar 3.7. Gambaran umum alur proses aplikasi monitoring .....</i>	<i>23</i>
<i>Gambar 3.8. Diagram use case aplikasi monitoring.....</i>	<i>23</i>
<i>Gambar 3.9. Rancangan antarmuka untuk fitur monitoring tangki .....</i>	<i>28</i>
<i>Gambar 3.10. Rancangan Antarmuka untuk fitur lihat riwayat monitoring tangki .....</i>	<i>29</i>
<i>Gambar 4.1. Fungsi setup pada child node.....</i>	<i>32</i>
<i>Gambar 4.2. Implementasi fungsi loop pada child node.....</i>	<i>33</i>
<i>Gambar 4.3. Fungsi read GPS .....</i>	<i>34</i>
<i>Gambar 4.4. Fungsi convert2degrees .....</i>	<i>35</i>
<i>Gambar 4.5. Fungsi getHeight .....</i>	<i>36</i>
<i>Gambar 4.6. Fungsi send_data .....</i>	<i>36</i>
<i>Gambar 4.7. Fungsi setup pada center node.....</i>	<i>37</i>
<i>Gambar 4.8. Fungsi loop pada center node .....</i>	<i>37</i>
<i>Gambar 4.9. Fungsi getVehicleStatus .....</i>	<i>38</i>

<i>Gambar 4.10. Fungsi calculateVolume .....</i>	<i>39</i>
<i>Gambar 4.11. Proses penggambaran icon pada peta dengan menggunakan jquery .....</i>	<i>39</i>
<i>Gambar 4.12. Halaman fitur monitoring tangki .....</i>	<i>40</i>
<i>Gambar 4.13. Halaman fitur mendaftarkan data tangki .....</i>	<i>40</i>
<i>Gambar 5.1. Prototipe tangki uji coba .....</i>	<i>46</i>
<i>Gambar 5.2. Kalibrasi sinyal pada saat booting .....</i>	<i>47</i>
<i>Gambar 5.3. Center node berhasil melakukan booting .....</i>	<i>48</i>
<i>Gambar 5.4. Lampu indikator pada Bluetooth .....</i>	<i>49</i>
<i>Gambar 5.5. Center node berhasil menerima data dari child node .....</i>	<i>50</i>
<i>Gambar 5.6. Bundle data yang dikirim ke server .....</i>	<i>50</i>
<i>Gambar 5.7. Data yang berhasil terkirim ke server disimpan pada tabel track_record .....</i>	<i>51</i>
<i>Gambar 5.8. Fitur monitoring tangki berhasil mengolah data yang ada pada tabel track_record .....</i>	<i>52</i>
<i>Gambar 5.9. Child node yang diberi pelampung sehingga bisa mengapung di air .....</i>	<i>53</i>
<i>Gambar 5.10. Center node .....</i>	<i>53</i>
<i>Gambar 5.11. Mendaftarkan data prototipe tangki .....</i>	<i>54</i>
<i>Gambar 5.12. Keadaan tangki untuk uji coba 1 .....</i>	<i>55</i>
<i>Gambar 5.13. Grafik akurasi sensor ultrasonic pada percobaan 1 .....</i>	<i>57</i>
<i>Gambar 5.14. Grafik akurasi volume cairan berdasarkan sensor ultrasonik pada percobaan 1 .....</i>	<i>57</i>
<i>Gambar 5.15. Keadaan tangki untuk skenario uji coba 2 .....</i>	<i>58</i>
<i>Gambar 5.16. Grafik akurasi sensor ultrasonic pada percobaan 2 .....</i>	<i>60</i>
<i>Gambar 5.17. Grafik akurasi volume cairan berdasarkan sensor ultrasonic pada percobaan 2 .....</i>	<i>60</i>
<i>Gambar 5.18. Prototipe tangki 3 akan dikurangi muatannya secara periodik .....</i>	<i>61</i>
<i>Gambar 5.19. Grafik kondisi tangki ketika muatan bocor / dicuri .....</i>	<i>62</i>
<i>Gambar 6.1. Modul Ethernet .....</i>	<i>66</i>

*Gambar 6.2. Arsitektur sistem jika menggunakan modul ethernet*  
.....66

# **BAB I**

## **PENDAHULUAN**

### **1.1 Latar Belakang**

Kasus pencurian minyak dengan cara mengambil sebagian muatan dari kapal tanker di Indonesia masih menjadi berita terkini. Salah satu contohnya adalah kasus pencurian minyak di Kepulauan Riau. Pada kasus tersebut, pelaku berhasil mendapatkan uang sebesar 1,3 triliun. Tindakan pencurian muatan kapal tanker ini dilakukan dengan cara “mengencingkan” muatan kapal tanker ke kapal ilegal. Oknum-oknum yang melakukan tindakan tidak terpuji tersebut kebanyakan berasal dari orang dalam (Pegawai Pertamina), sehingga membuat tindakan ini sulit dilacak. Dampak yang disebabkan oleh tindakan tersebut adalah kerugian yang berpengaruh pada perusahaan energi yang ada di Indonesia, dalam hal ini adalah Pertamina. Tindakan yang sebenarnya cukup mudah untuk dilakukan oleh semua orang, namun berdampak besar bagi kerugian Bangsa.

Sebenarnya untuk mengatasi masalah tersebut cukup mudah, karena saat ini cukup banyak alat-alat yang mempermudah pekerjaan manusia seperti sensor. Selain itu, sebagian besar masyarakat Indonesia sudah mulai memanfaatkan internet pada pekerjaannya, sehingga dengan memanfaatkan internet dan kemajuan teknologi saat ini, maka akan dibuat alat untuk memonitoring volume tangki dengan memanfaatkan sensor ultrasonik dan sensor lokasi. Sensor ultrasonik digunakan untuk mengukur *water level* atau kedalaman cairan yang ada pada tangki, sedangkan sensor lokasi digunakan untuk melacak posisi tangki. Namun untuk memonitoring data yang dihasilkan kedua sensor tersebut, maka dibutuhkan satu modul tambahan untuk mengirimkan data ke server. Dari server tersebut, data sensor akan diolah dan ditampilkan luarannya dalam bentuk aplikasi web. Dalam



aplikasi web tersebut akan ditampilkan peta lokasi tangki dan volume tangki tersebut.

Untuk mengatasi guncangan pada tangki agar data ketinggian cairan yang dikirim ke server oleh sensor ultrasonik sesuai, maka akan digunakan 2 sensor ultrasonik yang diletakan ujung tangki yang berbeda. Untuk sensor lokasi dan modul untuk mengirim data ke server, akan diletakan diluar tangki, sehingga diperlukan adanya komunikasi antar alat-alat tersebut. Maka dari itu, dalam tugas akhir ini, akan dibentuk jaringan sensor nirkabel yang menggunakan modul *Bluetooth* untuk integrasi antar node sensor. Dengan memanfaatkan jaringan sensor nirkabel tersebut, tentu data yang dikirim ke server sesuai dengan kondisi lingkungan saat itu, sehingga dengan data-data yang didapat dari sensor tersebut akan mempermudah monitoring keadaan tangki.

## 1.2 Rumusan Masalah

Rumusan masalah yang diangkat dalam tugas akhir ini dapat dipaparkan sebagai berikut:

1. Bagaimana merancang sebuah jaringan sensor nirkabel yang dapat memperoleh data sesuai dengan keadaan tangki?
2. Bagaimana membuat program untuk sensor yang dapat membungkus data dan mengirimkannya ke server?
3. Bagaimana membuat aplikasi yang dapat menampilkan peta lokasi dan kondisi tangki berdasarkan data yang diterima server?

## 1.3 Batasan Masalah

Permasalahan yang dibahas dalam tugas akhir ini memiliki beberapa batasan antara lain:

1. Aplikasi monitoring volume cairan tangki ini berbasis *web* dengan bahasa pemrograman PHP dengan *framework* Codeigniter dan *database* MySQL, serta memanfaatkan leaflet.js sebagai API untuk menampilkan peta dunia.

2. Untuk komunikasi data antar node sensor menggunakan modul *Bluetooth* yang dipasang pada masing-masing node.
3. Untuk tempat pemasangan sensor menggunakan mikrokontroler arduino.
4. Program yang dipasang pada mikrokontroler arduino menggunakan bahasa pemrograman Java.
5. Uji coba akan dilakukan pada *prototype* tangki yang berisi air biasa

#### **1.4 Tujuan**

Tujuan dari pembuatan tugas akhir ini antara lain :

1. Membuat aplikasi monitoring keadaan tangki berdasarkan sensor ultrasonik dan sensor lokasi.

#### **1.5 Manfaat**

Dengan dibuatnya Tugas Akhir ini, maka dapat mempermudah petugas pos keberangkatan atau kedatangan kendaraan dalam melacak posisi tangki jika terjadi kebocoran tangki atau pengambilan muatan tangki secara ilegal.

#### **1.6 Metodologi**

1. Penyusunan proposal tugas akhir

Proposal tugas akhir ini berisi tentang deskripsi pendahuluan dari tugas akhir yang akan dibuat. Pendahuluan pada proposal tugas akhir ini terdiri dari latar belakang diajukannya usulan tugas akhir, rumusan masalah yang diangkat, batasan masalah untuk tugas akhir, tujuan dari pembuatan tugas akhir, dan manfaat hasil dari pembuatan tugas akhir. Selain itu dijelaskan pula tinjauan pustaka yang digunakan sebagai referensi pendukung implementasi tugas akhir. Pada proposal ini juga terdapat perencanaan jadwal pengerjaan tugas akhir.

2. Studi literatur

Pada studi literatur ini, akan dipelajari sejumlah referensi yang diperlukan dalam implementasi sistem, yaitu mengenai

mikrokontroler Arduino, sensor ultrasonik, GPS (Global Positioning System) dan Modul GPS/GSM/GPRS Versi 3.0.

3. Analisis dan desain perangkat lunak

Tahap ini meliputi perancangan sitem berdasarkan studi literature dan pembelajaran konsep teknologi dari perangkat lunak yang ada. Langkah-langkah yang dikerjakan juga didefinisikan pada tahap ini. Pada tahapan ini dibuat prototype sistem, yang merupakan rancangan dasar dari sistem yang akan dibuat. Serta dilakukan desain suatu sistem dan desain proses-proses yang ada.

4. Implementasi perangkat lunak

Implementasi merupakan tahap membangun rancangan program yang telah dibuat. Pada tahapan ini merealisasikan apa yang terdapat pada tahapan sebelumnya, sehingga menjadi sebuah program yang sesuai dengan apa yang telah direncanakan.

5. Pengujian dan evaluasi

Pada tahapan ini dilakukan uji coba pada alat yang telah dirancang. Tahapan ini dimaksudkan untuk mengevaluasi tingkat akurasi dari alat tersebut serta mencari masalah yang mungkin timbul dan mengadakan perbaikan jika terdapat kesalahan.

6. Penyusunan Buku Tugas Akhir

Pada tahap ini dilakukan penyusunan laporan yang menjelaskan dasar teori dan metode yang digunakan dalam tugas akhir ini serta hasil dari implementasi aplikasi perangkat lunak yang telah dibuat

## **1.7 Sistematika Penulisan Laporan Tugas Akhir**

Buku Tugas Akhir ini bertujuan untuk mendapatkan gambaran dari pengerjaan Tugas Akhir ini. Selain itu, diharapkan dapat berguna untuk pembaca yang tertarik untuk melakukan

pengembangan lebih lanjut. Secara garis besar, buku Tugas Akhir terdiri atas beberapa bagian seperti berikut ini:

**Bab I      Pendahuluan**

Bab yang berisi mengenai latar belakang, tujuan, dan manfaat dari pembuatan Tugas Akhir. Selain itu permasalahan, batasan masalah, metodologi yang digunakan, dan sistematika penulisan juga merupakan bagian dari bab ini.

**Bab II     Dasar Teori**

Bab ini berisi penjelasan secara detail mengenai dasar-dasar penunjang dan teori-teori yang digunakan untuk mendukung pembuatan Tugas Akhir ini.

**Bab III    Perancangan Perangkat Lunak**

Bab ini berisi tentang desain sistem yang disajikan dalam bentuk *pseudocode*.

**Bab IV    Implementasi**

Bab ini membahas implementasi dari desain yang telah dibuat pada bab sebelumnya. Penjelasan berupa code yang digunakan untuk proses implementasi.

**Bab V     Uji Coba Dan Evaluasi**

Bab ini menjelaskan kemampuan perangkat lunak dengan melakukan pengujian kebenaran dan pengujian kinerja dari sistem yang telah dibuat.

**Bab VI    Kesimpulan Dan Saran**

Bab ini merupakan bab terakhir yang menyampaikan kesimpulan dari hasil uji coba yang dilakukan dan saran untuk pengembangan perangkat lunak ke depannya.

## **BAB II**

### **TINJAUAN PUSTAKA**

Bab ini berisi penjelasan teori-teori yang berkaitan dengan rancangan alat yang diajukan pada pengimplementasian program. Penjelasan ini bertujuan untuk memberikan gambaran secara umum terhadap alat yang dirancang dan berguna sebagai penunjang dalam pengembangan perangkat lunak.

#### **2.1 Mikrokontroler Arduino**

Arduino adalah kit elektronik atau papan rangkaian elektronik *open source* yang di dalamnya terdapat komponen utama yaitu sebuah chip mikrokontroler dengan jenis AVR dari perusahaan Atmel. Mikrokontroler itu sendiri adalah chip atau IC (integrated circuit) yang bisa diprogram menggunakan komputer.



***Gambar2.1. Mikrokontroler Arduino***

Tujuan menanamkan program pada mikrokontroler adalah agar rangkaian elektronik dapat membaca input, memproses input

tersebut dan kemudian menghasilkan output sesuai yang diinginkan. Jadi mikrokontroler bertugas sebagai ‘otak’ yang mengendalikan input, proses dan output sebuah rangkaian elektronik. Mikrokontroler ada pada perangkat elektronik di sekeliling kita. Misalnya handphone, MP3 player, DVD, televisi, AC, dll. Mikrokontroler juga dipakai untuk keperluan mengendalikan robot. Baik robot mainan, maupun robot industri. Karena komponen utama Arduino adalah mikrokontroler, maka Arduino pun dapat diprogram menggunakan komputer sesuai kebutuhan.

Arduino memiliki beberapa tipe berdasarkan kegunaan dan memori yang ada pada mikrokontroler tersebut. Berikut adalah contoh-contoh arduino:

1. Arduino Uno. Jenis yang ini adalah yang paling banyak digunakan. Terutama untuk pemula sangat disarankan untuk menggunakan Arduino Uno. Dan banyak sekali referensi yang membahas Arduino Uno. Versi yang terakhir adalah Arduino Uno R3 (Revisi 3), menggunakan ATMEGA328 sebagai Microcontrollernya, memiliki 14 pin I/O digital dan 6 pin input analog. Untuk pemograman cukup menggunakan koneksi USB type A to To type B. Sama seperti yang digunakan pada USB printer.



***Gambar 2.2. Arduino UNO***

2. Arduino Due. Berbeda dengan saudaranya, Arduino Due tidak menggunakan ATMEGA, melainkan dengan chip yang lebih tinggi ARM Cortex CPU. Memiliki 54 I/O pin digital dan 12 pin input analog. Untuk pemrogramannya menggunakan Micro USB, terdapat pada beberapa handphone.



***Gambar 2.3. Arduino Due***

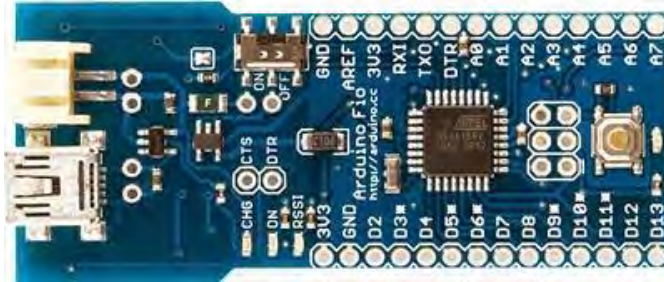
3. Arduino Leonardo. Bisa dibilang Leonardo adalah saudara kembar dari Uno. Dari mulai jumlah pin I/O digital dan pin input Analognya sama. Hanya pada Leonardo menggunakan Micro USB untuk pemrogramannya.



***Gambar 2.4. Arduino Leonardo***

4. Arduino Fio. Bentuknya lebih unik, terutama untuk socketnya. Walau jumlah pin I/O digital dan input analognya sama dengan uno dan leonardo, tapi Fio memiliki Socket XBee. XBee

membuat Fio dapat dipakai untuk keperluan proyek yang berhubungan dengan wireless



*Gambar 2.5. Arduino Fio*

5. Arduino Mega. Mirip dengan Arduino Uno, sama-sama menggunakan USB type A to B untuk pemrogramannya. Tetapi Arduino Mega, menggunakan Chip yang lebih tinggi ATMEGA2560. Dan tentu saja untuk Pin I/O Digital dan pin input Analognya lebih banyak dari Uno.



*Gambar 2.6. Arduino Mega*



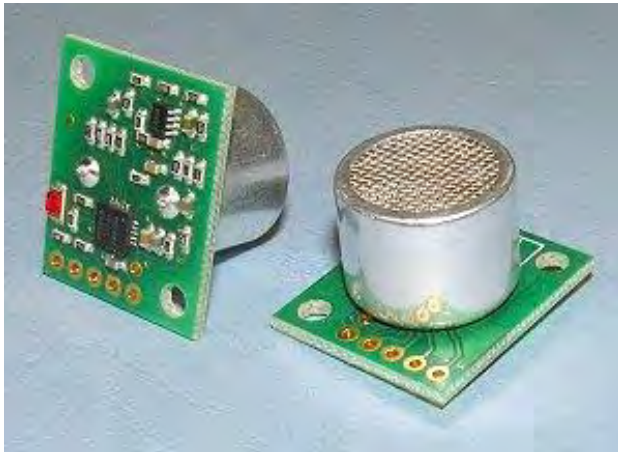
## 2.2 Sensor Ultrasonik

Sensor Ultrasonik adalah alat elektronika yang kemampuannya bisa mengubah dari energi listrik menjadi energi mekanik dalam bentuk gelombang suara ultrasonik. Sensor ini terdiri dari rangkaian pemancar Ultrasonik yang dinamakan *transmitter* dan penerima ultrasonik yang disebut *receiver*. Alat ini digunakan untuk mengukur gelombang ultrasonik. Gelombang ultrasonik adalah gelombang mekanik yang memiliki ciri-ciri longitudinal dan biasanya memiliki frekuensi di atas 20 Khz. Gelombang Ultrasonik dapat merambat melalui zat padat, cair maupun gas. Gelombang Ultrasonik adalah gelombang rambatan energi dan momentum mekanik sehingga merambat melalui ketiga element tersebut sebagai interaksi dengan molekul dan sifat enersia medium yang dilaluinya.

Ada beberapa penjelasan mengenai gelombang ultrasonik. Sifat dari gelombang ultrasonik yang melalui medium menyebabkan getaran partikel dengan medium aplitudo sama dengan arah rambat longitudinal sehingga menghasilkan partikel medium yang membentuk suatu rapatan atau biasa disebut Strain dan tegangan yang biasa disebut Strees. Proses lanjut yang menyebabkan terjadinya rapatan dan regangan di dalam medium disebabkan oleh getaran partikel secara periodik selama gelombang ultrasonik lainnya. Gelombang ultrasonik merambat melalui udara dengan kecepatan 344 meter per detik, mengenai objek dan memantul kembali ke *sensor ultrasonik*. Seperti yang telah umum diketahui, gelombang ultrasonik hanya bisa didengar oleh makhluk tertentu seperti kelelawar dan ikan paus. Kelelawar menggunakan gelombang ultrasonic untuk berburu di malam hari sementara paus menggunakannya untuk berenang di kedalaman laut yang gelap.

Perhitungan waktu yang diperlukan modul sensor Ping untuk menerima pantulan pada jarak tertentu mempunyai rumus  $S = (tIN \times V) : 2$ . Rumus tersebut mempunyai keterangan sebagai berikut. ( $S$ ) adalah jarak antara sensor ultrasonik dengan objek yang terdeteksi. ( $V$ ) adalah cepat rambat gelombang ultrasonik di udara dengan kecepatan normal (344 meter per detik) ( $tIN$ ) adalah selisih

waktu pemancaran dan penerimaan pantulan gelombang. Ada 3 prinsip kerja dari sensor ultrasonik yaitu, sinyal dipancarkan melalui pemancar gelombang ultrasonik. Sinyal yang dipancarkan akan merambat sebagai gelombang bunyi dengan kecepatan bunyi berkisar 344 m/s. Dan yang terakhir sinyal yang sudah diterima akan diproses untuk menghitung jaraknya.



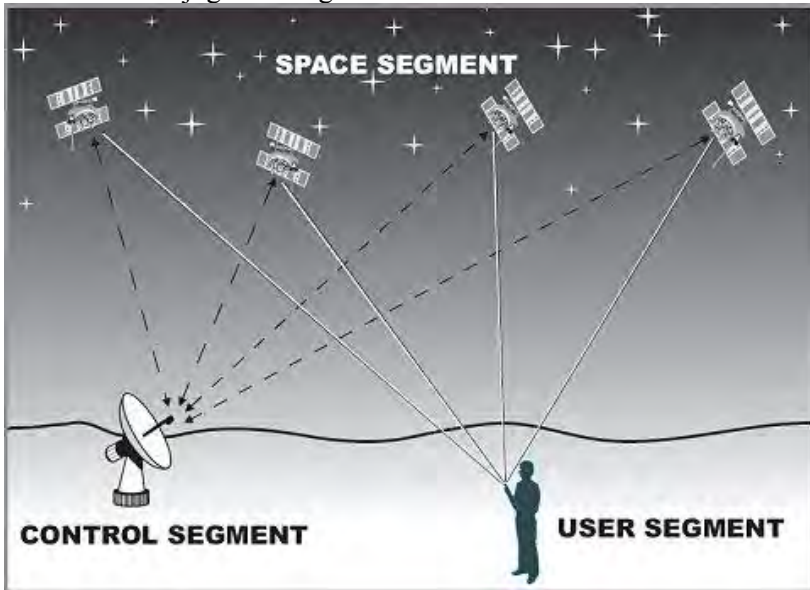
***Gambar 2.7. Sensor Ultrasonik***

### **2.3 GPS (*Global Positioning System*)**

GPS (*Global Positioning System*) adalah sebuah sistem navigasi berbasis radio yang menyediakan informasi koordinat posisi, kecepatan, dan waktu kepada pengguna di seluruh dunia. Jasa penggunaan satelit GPS tidak dikenakan biaya. Pengguna hanya membutuhkan GPS receiver untuk dapat mengetahui koordinat lokasi. Keakuratan koordinat lokasi tergantung pada tipe GPS receiver.

GPS terdiri dari tiga bagian yaitu satelit yang mengorbit bumi (Satelit GPS mengelilingi bumi 2x sehari), stasiun pengendali dan pemantau di bumi, dan GPS receiver (alat penerima GPS).

Satelit GPS dikelola oleh Amerika Serikat. Alat penerima GPS inilah yang dipakai oleh pengguna untuk melihat koordinat posisi. Selain itu GPS juga berfungsi untuk menentukan waktu.



**Gambar 2.8. Butuh minimal 3 satelit untuk mendapatkan koordinat fix dari GPS**

Format data keluaran GPS ditetapkan oleh NMEA (*National Marine Electronics Assosiation*). Data keluaran dalam format NMEA berbentuk kalimat (*string*) yang merupakan rangkaian karakter ASCII 8 bit. Setiap kalimat diawali dengan satu karakter '\$', dua karakter *talker ID*, tiga karakter *sentence ID* dan diikuti oleh *data fields* yang masing-masing dipisahkan oleh koma serta diakhiri oleh *optional checksum*. Contoh data GPS yang didapat seperti '\$GPGGA,061648,0.276513,S,112.791692,E,1,09,0.8,70.0,M,1.5,M,\*,5E'. Penjelasan dari contoh data tersebut adalah sebagai berikut.

- 'GPGGA' = *sentence identifier*.
- '061648' = waktu.

- ‘0.276513’ = *latitude*.
- ‘S’ = indikator (S = *South*, N = *North*).
- ‘112.791692’ = *longitude*.
- ‘E’ = indikator (E = *East*, W = *West*).
- ‘1’ = posisi (0 = *invalid*, 1 = *valid*).
- ‘09’ = jumlah satelit.
- ‘0.8’ = presisi secara horizontal.
- ‘70 M’ = *altitude*.
- ‘1.5 M’ = ketinggian.
- ‘5E’ = *checksum*.

## 2.4 GPS/GSM/GPRS Shield Versi 3.0

GPS/GSM/GPRS *Shield* versi 3.0 merupakan *shield* produksi DFRobot. *Shield* ini mendukung frekuensi Quad-band GSM/GPRS pada 850 MHz, 900 MHz, 1800 MHz, dan 1900 MHz. *Shield* ini juga dilengkapi dengan antena GPS untuk keperluan navigasi satelit sehingga memungkinkan sebuah robot atau sistem mengirim data lokasi melalui jaringan GSM. *Shield* ini juga mendukung format data GPS dengan protokol NMEA [1].

Dalam penggunaan *shield* ini dibutuhkan *command* khusus yang disebut *AT command*. *AT command* akan diprogram dan *shield* akan membaca perintah sesuai dengan *command* dalam program. *AT command* sendiri memiliki format penulisan yang sudah diatur dari pabrik pembuat *shield*. Beberapa contoh *AT command* yang digunakan pada sistem ini ini dapat dilihat pada Tabel 2.1.

**Tabel 2.1. AT Command pada GPS/GPRS/GSM Shield V3.0**

No.	AT Command	Keterangan
1	AT+CGSPWR	Menyalakan GPS
2	AT+CGPSRST	Mengatur Mode GPS
3	AT+CGPSINF	Mendapatkan informasi lokasi

4	AT+CGREG=?	Mengecek staus jaringan
5	AT+HTTPIPINIT	Inisiasi mode HTTP
6	AT+HTTPTERM	Menyudahi mode HTTP
7	AT+HTTPPARAM	Mengatur URL yang akan dikunjungi

## 2.5 Modul *Bluetooth* HC 05 & HC 06

Modul *bluetooth* HC 05 dan HC 06 merupakan modul *Bluetooth* yang kompatibel dengan Arduino. Modul ini berfungsi sebagai *plugin* pada arduino yang bertugas sebagai penerima atau pengirim data melalui media udara (*wireless*). Perbedaan dari HC 05 dan HC 06 adalah, pada modul HC 05 terdapat perintah untuk mengatur mode dari *Bluetooth*, sehingga HC 05 dapat berperan sebagai *slave* atau *master*, sedangkan pada modul HC 06 tidak terdapat perintah untuk mengatur mode *Bluetooth* sehingga HC 06 hanya bias berperan sebagai *master* saja. Pada kedua modul ini juga digunakan *AT command* sebagai bahasa untuk menjalankan perintah tertentu. Berapa contoh *AT command* yang digunakan dalam modul *bluetooth* HC 05 dan HC 06 dapat dilihat pada tabel 2.2.

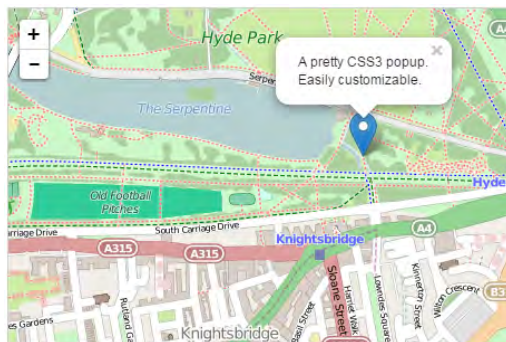
**Tabel 2.2. *AT Command* pada *Bluetooth* HC 05 dan HC 06**

No.	<i>AT Command</i>	Keterangan	HC 05	HC 06
1	AT+RESET	Mereset modul	V	-
2	AT+RMAAD	Memutuskan koneksi pada perangkat yang sedang tersambung	V	-
3	AT+INIT	Memulai modul	V	-
4	AT+LINK	Menyambung ke perangkat lain dengan parameter alamat MAC	V	-
5	AT+ROLE	Mengatur mode ( <i>slave</i> / <i>master</i> )	V	-

6	AT+NAME	Mengganti perangkat	nama	V	V
---	---------	---------------------	------	---	---

## 2.6 Leaflet.js API

*Leaflet* merupakan API untuk penggambaran peta dunia interaktif berbasis *javascript open-source*. Dikembangkan oleh Vladimir Agafonkin bersama tim kontributor. Dengan hanya memiliki besar 33 KB, *leaflet* sudah memiliki semua fitur yang diperlukan oleh pengembang aplikasi yang memerlukan API peta dunia. *Leaflet* dirancang dengan mengutamakan kesederhanaan, performa dan kegunaan. *Leaflet* mampu bekerja secara efisien pada semua desktop maupun aplikasi *mobile*, dengan memanfaatkan HTML5 dan CSS3. API ini dapat dikembangkan dengan memanfaatkan banyak *plugin* yang tersedia dan memiliki dokumentasi yang bagus, sederhana, dan memiliki kode yang mudah untuk dipelajari sehingga membuat para kontributor senang untuk mengembangkan API ini. API mengambil gambar masing-masing *tile* peta dunia dari server *openstreetmap*.



**Gambar 2.9. Penggambaran peta dunia dengan menggunakan API leaflet.js**

## **BAB III**

### **DESAIN PERANGKAT**

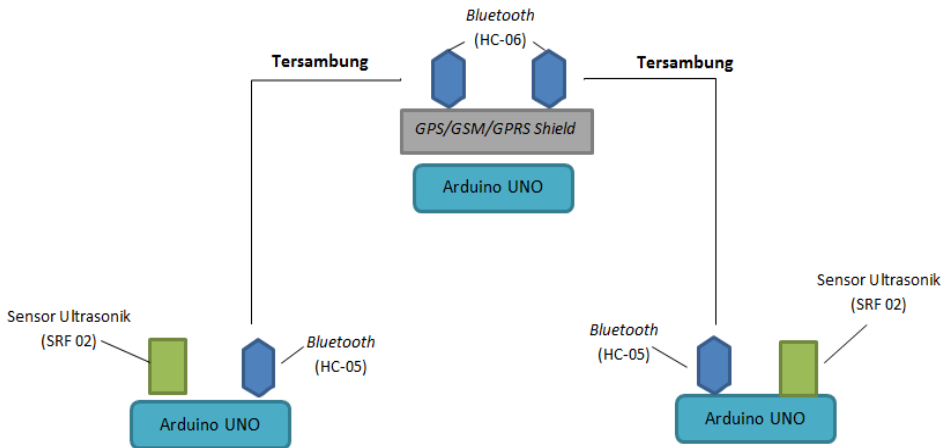
Pada bab ini akan dijelaskan perancangan alat dan program yang dibuat. Perancangan akan dibagi menjadi dua proses utama, yaitu:

1. Perancangan mikrokontroler Arduino dengan integrasi sensor ultrasonik untuk menangkap data kedalaman cairan pada tangki serta data lokasi tangki.
2. Perancangan aplikasi monitoring untuk memonitor keadaan tangki yang dipasang mikrokontroler.

Pada bab ini juga akan dijelaskan mengenai gambaran dan alur kerja sistem secara umum.

#### **3.1 Perancangan Integrasi Mikrokontroler dan Komponen Sensor**

Sensor ultrasonik merupakan salah satu alat utama pada sistem. Sensor tersebut akan digunakan untuk menangkap data kedalaman cairan pada tangki. Untuk menambah tingkat kevalidan data, maka tidak hanya menggunakan satu sensor ultrasonik saja, namun menggunakan dua sensor ultrasonik, sehingga diperlukan integrasi antar alat tersebut. Untuk menangkap data lokasi dan mengirim data ke server, maka diperlukan modul tambahan yaitu modul *GPS/GSM/GPRS Shield*. Agar alat-alat pada sistem dapat saling berintegrasi, maka akan dibentuk jaringan kecil berbasis nirkabel, dalam hal ini menggunakan modul *Bluetooth* sebagai media komunikasi antar *node*. Gambaran umum perancangan alat dapat dilihat pada Gambar 3.1.



**Gambar 3.1. Rancangan integrasi alat secara umum**

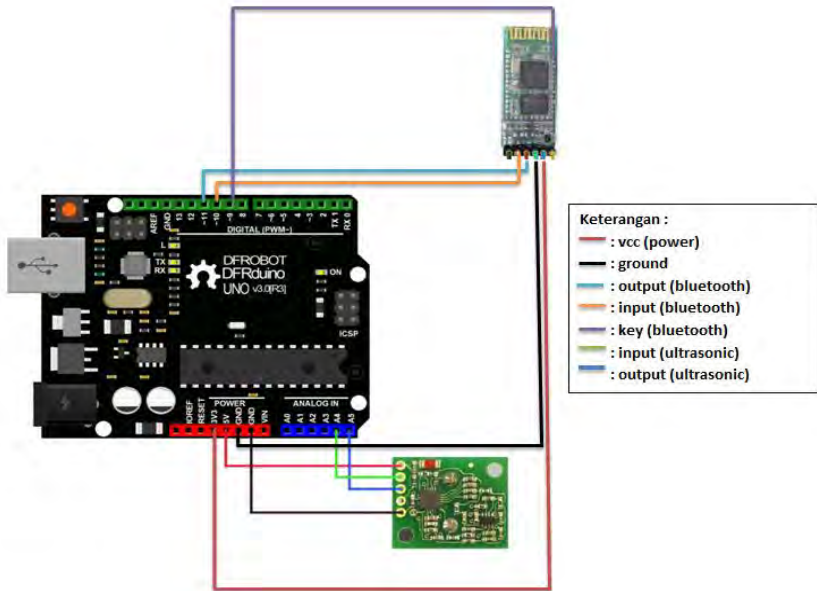
Pada gambar tersebut, mikrokontroler yang dipasang modul *GPS/GSM/GPRS Shield* merupakan node utama dalam jaringan, karena node tersebut yang bertugas menerima dan mengirim data, sehingga untuk selanjutnya node tersebut dinamakan *center node*. Sedangkan untuk mikrokontroler yang dipasang sensor ultrasonik merupakan node *slave* karena node tersebut hanya bertugas untuk mengirim data ke *center node*, sehingga untuk selanjutnya node tersebut dinamakan *child node*.

### 3.1.1 Perancangan *Child Node*

Seperti pada Gambar 3.1, *child node* terdiri dari mikrokontroler Arduino, sensor ultrasonik, dan *Bluetooth*. Sensor ultrasonik yang akan digunakan sensor ultrasonic seri SRF 02 yang memiliki kemampuan untuk mendeteksi jarak dari 16 cm hingga 6m.



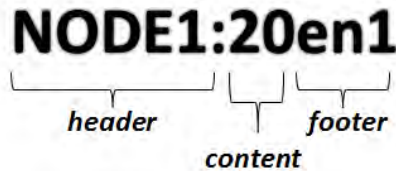
*Bluetooth* yang akan digunakan adalah *Bluetooth* seri HC-05 yang memiliki kemampuan *scan device* pada jarak maksimal 9 meter. Kedua alat tersebut akan dipasangkan ke mikrokontroler Arduino, sehingga dapat dikendalikan melalui program. Perancangan alat pada *child node* dapat dilihat pada Gambar 3.2.



**Gambar 3.2. Perancangan Child node**

Mikrokontroler Arduino memerlukan program untuk mengendalikan jalannya mikrokontroler tersebut. Program yang akan dirancang adalah program yang dapat menerima data dari sensor ultrasonik dan mengirimkannya ke *center node* melalui media *Bluetooth* dalam interval waktu setiap 10 detik. Data jarak yang diterima dari sensor ultrasonik adalah dalam bentuk integer (dalam satuan sentimeter). Untuk mengirim data tersebut melalui *Bluetooth*, perlu dibuat protokol sederhana agar data yang dikirim ke *center node* dapat diolah dengan valid. Protokol tersebut terdiri dari

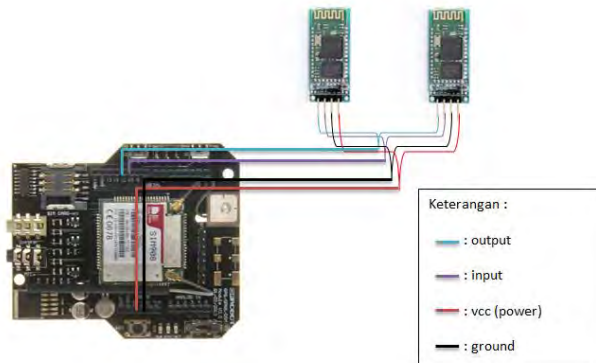
*header*, *content*, dan *footer*. *Header* sendiri berisi informasi asal pesan, *content* berisi isi pesan, dan *footer* berisi akhir dari pesan tersebut. Contoh protokol tersebut dapat dilihat pada Gambar 3.3.



**Gambar 3.3. Protokol pengiriman dari child node**

### 3.1.2 Perancangan Center Node

*Center node* merupakan node utama pada jaringan sensor tersebut. Node ini bertugas untuk menerima dan mengirimkan data sensor ke server. *Center node* terdiri dari mikrokontroler arduino, *GPS/GSM/GPRS Shield*, dan modul *Bluetooth*.

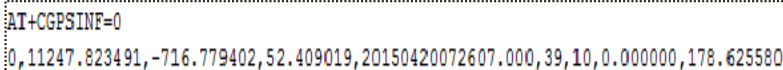


**Gambar 3.4. Perancangan Center Node**

Pada *center node*, seri *Bluetooth* yang digunakan adalah seri HC 06 yang hanya dapat bertugas sebagai master, yakni hanya bisa

menerima data dari *Bluetooth* lain. Pada *center node* diperlukan 2 modul *Bluetooth*, karena metode komunikasi *Bluetooth* adalah *one-to-one connection*, artinya satu *Bluetooth* hanya bisa melakukan koneksi ke satu *device* dalam satu waktu.

*GPS/GSM/GPRS Shield* merupakan komponen utama pada *center node*, karena modul tersebut yang akan menangkap data GPS dari satelit, dan mengirimkan data GPS dan data dari *child node* ke server. Modul ini mengirimkan data ke server dengan cara memanggil *web service* yang disediakan oleh aplikasi server. Data GPS yang diterima dari satelit merupakan data mentah yang harus diolah agar didapatkan data *latitude* dan *longitude* yang valid. Contoh data mentah yang diterima oleh modul tersebut dapat dilihat pada gambar 3.5.

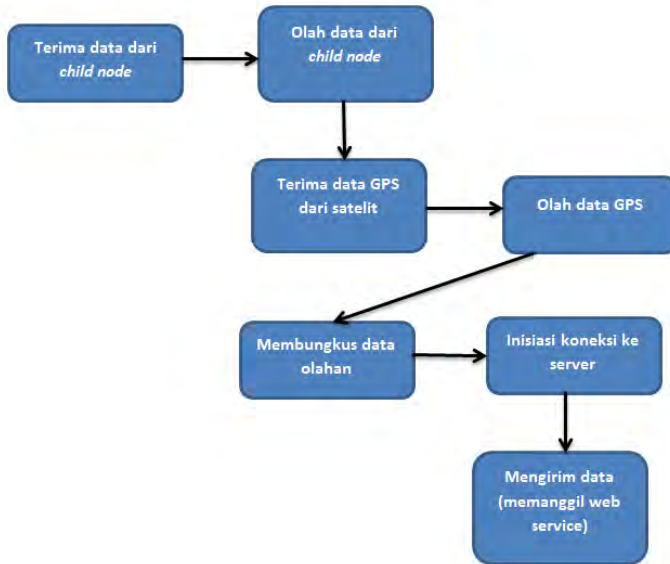


```
AT+CGPSINF=0
0,11247.823491,-716.779402,52.409019,20150420072607.000,39,10,0.000000,178.625580
```

**Gambar 3.5. Data mentah yang diterima oleh modul GPS/GSM/GPRS Shield**

Kedua jenis komponen tersebut (*bluetooth* dan *GPS/GSM/GPRS Shield*) akan di pasangkan ke mikrokontroler arduino sehingga dapat dikendalikan melalui program. Perancangan alat pada *center node* dapat dilihat pada gambar 3.4.

Seperti halnya *child node*, *center node* juga memerlukan program untuk mengendalikan jalannya komponen. Program yang dirancang untuk *center node* adalah program yang dapat menerima dan mengolah data dari GPS maupun dari *child node* dan membungkus kedua data olahan tersebut, lalu melakukan inisiasi koneksi ke server sehingga *center node* dapat mengirimkan data tersebut dengan cara memanggil *web service* yang disediakan oleh server. Alur proses dari *center node* dapat dilihat pada Gambar 3.6.



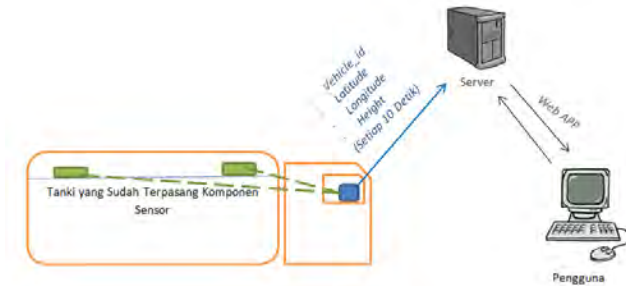
**Gambar 3.6. Alur proses dari center node**

*Center node* tidak akan mengirim data ke *server* sebelum menerima data dari kedua *child node*, karena data yang akan dikirim ke *server* merupakan data rata-rata dari kedua *childnode*. *Center node* akan mengirimkan data ke *server* begitu data dari kedua *child node* diterima, sehingga interval waktu pengiriman data dari *center node* ke *server* adalah setiap 10 detik.

### 3.2 Perancangan Aplikasi Monitoring

Untuk mengolah data keluaran dari komponen yang telah dirancang (*child node* & *center node*), diperlukan sebuah aplikasi yang dapat memudahkan pengguna dalam membaca data olahan. Aplikasi yang dirancang untuk sistem ini merupakan aplikasi berbasis *web*, sehingga dapat aplikasi ini dapat dibuka dimana saja

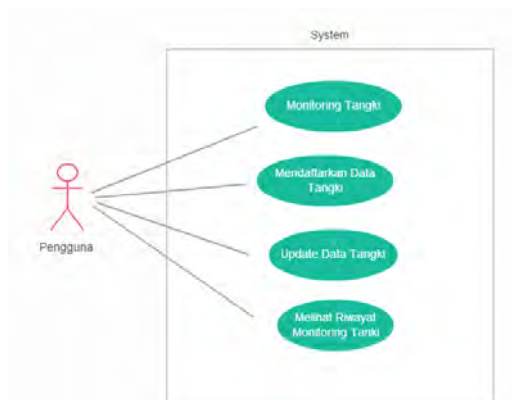
dan kapan saja. Gambaran umum alur proses aplikasi monitoring dapat dilihat pada gambar 3.7.



**Gambar 3.7. Gambaran umum alur proses aplikasi monitoring**

### 3.2.1 Perancangan Diagram Use Case

Diagram *use case* untuk aplikasi monitoring dapat dilihat pada gambar 3.8.



**Gambar 3.8. Diagram use case aplikasi monitoring**

Dalam aplikasi monitoring tersebut, pengguna dapat melakukan 4 hal inti, yang pertama adalah pengguna dapat memonitoring tangki yang telah dipasang komponen jaringan sensor. Pengguna dapat melihat pergerakan dari kendaraan yang membawa tangki tersebut dan persentase volume isi tangki pada kendaraan. Kedua, pengguna dapat mendaftarkan data kendaraan tangki yang akan dimonitoring, dalam hal ini adalah kendaraan tangki yang dipasang komponen jaringan sensor. Ketiga, pengguna dapat memperbarui data kendaraan tangki yang telah didaftarkan sebelumnya. Data tangki yang dapat diperbarui adalah data tangki kendaraan yang tidak sedang dalam perjalanan, jadi data tangki yang dapat diperbarui adalah data tangki kendaraan yang belum berangkat atau yang sudah sampai tujuan. Keempat, pengguna dapat melihat riwayat dari tangki yang sudah dimonitoring sebelumnya.

***Tabel 3.1. Perancangan tabel vehicle\_master***

No.	Nama Atribut	Tipe Data	Keterangan
1	<i>vehicle_id</i>	<i>varchar</i>	Sebagai <i>primary key</i> untuk kendaraan
2	<i>vehicle_name</i>	<i>varchar</i>	Nama kendaraan
3	<i>tank_type</i>	<i>varchar</i>	Tipe tangki, digunakan untuk menentukan rumus untuk menghitung volume tangki
4	<i>diameter</i>	<i>double</i>	Diameter dari tangki, namun jika tipe tangki merupakan persegi panjang, diameter akan diisi dengan lebar tangki
5	<i>longsize</i>	<i>double</i>	Panjang tangki
6	<i>tank_height</i>	<i>double</i>	Tinggi tangki

### **3.2.2 Perancangan Basis Data**

Pada perancangan basis data untuk aplikasi monitoring, diperlukan 2 tabel master, yang pertama adalah table untuk menyimpan data tangki yang akan dinamakan tabel *vehicle\_master*,

yang kedua adalah tabel *user* yang digunakan untuk menyimpan akun pengguna, sehingga tidak sembarang orang dapat mengakses aplikasi monitoring. Selain itu juga dibutuhkan 2 tabel transaksi, yaitu tabel yang digunakan untuk menyimpan data yang diterima dari komponen sensor, yang akan dinamakan tabel *track\_record*, dan yang kedua adalah tabel yang digunakan untuk menyimpan data awal kondisi tangki sebelum berangkat, yang akan dinamakan *initial\_data*. Perancangan tabel pada database dapat dilihat pada Tabel 3.1, Tabel 3.2, Tabel 3.3, dan Tabel 3.4.

***Tabel 3.2. Perancangan tabel user***

No.	Nama Attribut	Tipe Data	Keterangan
1	<i>username</i>	<i>Varchar</i>	Sebagai <i>primary key</i> untuk akun
2	<i>password</i>	<i>varchar</i>	password yang disimpan merupakan <i>hash</i> dari password yang asli

***Tabel 3.3. Perancangan tabel initial\_data***

No.	Nama Attribut	Tipe Data	Keterangan
1	<i>Id</i>	<i>integer</i>	<i>autoincrement</i>
2	<i>vehicle id</i>	<i>varchar</i>	Sebagai <i>ID</i> dari kendaraan
3	<i>initial_volume</i>	<i>double</i>	Volume awal tangki sebelum berangkat
4	<i>is_going</i>	<i>integer</i>	<i>Flag</i> untuk menandakan kendaraan telah berangkat atau belum
5	<i>is_arrive</i>	<i>integer</i>	<i>Flag</i> untuk menandakan kendaraan telah sampai tujuan atau belum

**Tabel 3.4. Perancangan tabel *track\_record***

No.	Nama Atribut	Tipe Data	Keterangan
1	<i>id</i>	<i>integer</i>	<i>autoincrement</i>
2	<i>vehicle id</i>	<i>varchar</i>	Sebagai <i>ID</i> dari kendaraan
3	<i>latitude</i>	<i>double</i>	Koordinat <i>latitude</i> kendaraan
4	<i>Longitude</i>	<i>double</i>	Koordinat <i>longitude</i> kendaraan
5	<i>liquid level</i>	<i>double</i>	Ketinggian cairan didalam tangki
6	<i>Timestamp</i>	<i>datetime</i>	Waktu dari data yang masuk

### 3.2.3 Perancangan *Web Service*

*Web service* pada aplikasi monitoring digunakan untuk menerima data dari komponen sensor pada tangki. *Web service* yang dirancang merupakan *web service* sederhana yang memanfaatkan metode *GET* pada *http*, sehingga komponen sensor hanya perlu memanggil *url* dari *web service* ini dan menambahkan parameter-parameter. Parameter yang ditangkap oleh *web service* ini antara lain *ID* kendaraan (*vehicle\_id*), data ketinggian dari sensor ultrasonik (*height*), data koordinat *latitude* (*latitude*), dan data koordinat *longitude* (*longitude*). Selanjutnya data tersebut akan disimpan didalam database pada tabel *track\_record*.

### 3.2.4 Fungsi *getVehicleStatus*

Fungsi ini merupakan fungsi utama dari aplikasi monitoring. Fungsi inilah yang akan mengolah data dari komponen sensor yang telah dikirim ke server dan menampilkannya ke aplikasi, sehingga mudah dibaca oleh pengguna. Fungsi ini membaca data dari tabel *track\_record* dan tabel *initial\_data*, untuk membandingkan hasil perhitungan volume dari tinggi cairan yang didapat dari komponen sensor dengan volume awal pada saat kendaraan belum berangkat. Untuk perhitungan volume tangki berdasarkan ketinggian cairan

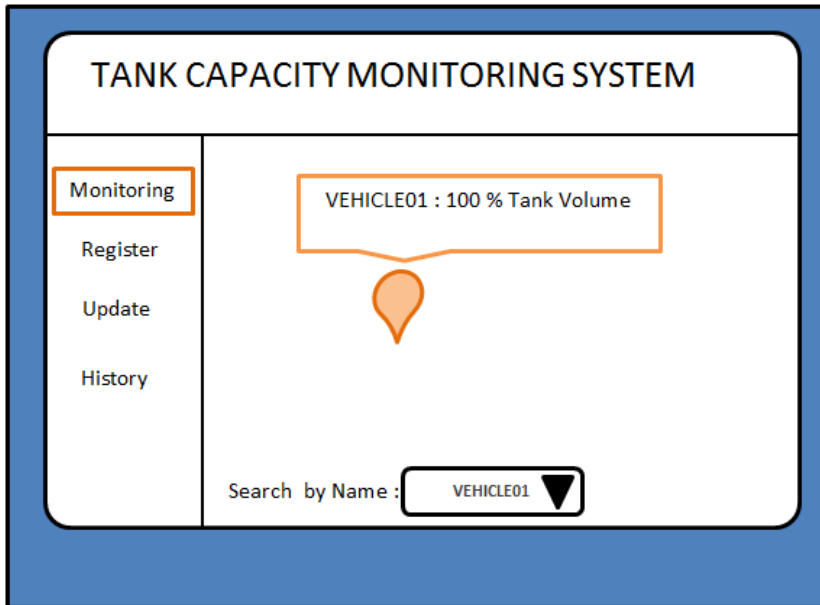


yang didapat dari komponen sensor, akan menggunakan rumus yang sesuai dengan tipe tangki. Jika tipe tangki berupa balok, maka perhitungan volume sesuai dengan rumus untuk menghitung volume balok, berdasarkan lebar dan panjang tangki yang ada pada tabel *vehicle\_master*, serta berdasarkan data tinggi yang diterima dari komponen sensor. Rumus volume yang ada pada aplikasi hanya ada 2, yaitu rumus untuk menghitung volume balok dan rumus untuk menghitung volume tabung yang memiliki 2 diameter (jenis tangki elips), karena kedua bentuk tersebut merupakan bentuk dari kebanyakan tangki yang digunakan.

Data yang diolah tadi, termasuk hasil dari perhitungan volume tangki, akan dikirimkan dalam bentuk *JSON (Javascript Object Notation)* dari *controller* ke *view*, sehingga data tersebut dapat ditampilkan pada halaman monitoring yang memanfaatkan *API* dari *leaflet.js* yang berfungsi untuk menampilkan peta dunia. Pada halaman monitoring tersebut, data tangki akan ditampilkan dalam bentuk persentase hasil dari perbandingan antara perhitungan volume tangki berdasarkan data ketinggian dari komponen sensor dengan volume tangki yang disimpan sebelum kendaraan berangkat.

### 3.2.5 Perancangan Antarmuka Sistem

Sesuai dengan perancangan pada diagram *use case*, pengguna dapat menjalankan 4 fitur utama, yaitu monitoring tangki, mendaftarkan data tangki, memperbarui data tangki, dan melihat riwayat tangki. Masing-masing fitur tersebut akan dirancang antar muka yang sederhana dan mudah dipahami oleh pengguna. Terdapat 1 menu tambahan yaitu menu *arrival*, untuk menandai kendaraan tangki yang sudah sampai tujuan pengiriman. Kendaraan tangki yang sudah ditandai atau sudah sampai di tujuan pengiriman tidak akan tampil di layar monitoring, namun tampil di halaman melihat riwayat tangki. Rancangan antarmuka untuk fitur monitoring tangki dapat dilihat pada Gambar 3.9.

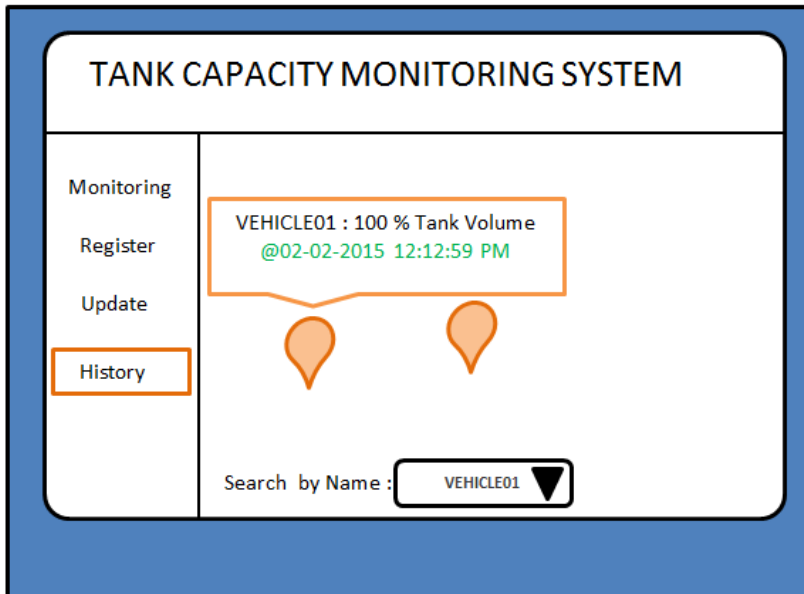


**Gambar 3.9. Rancangan antarmuka untuk fitur monitoring tangki**

Pada rancangan antarmuka tersebut terdapat *dropdown list* untuk mempermudah pengguna dalam memilih kendaraan mana yang akan dimonitoring. Rancangan antarmuka tersebut memanfaatkan API *leaflet.js* untuk menampilkan peta dunia sesuai dengan koordinat yang diterima dari fungsi *getVehicleStatus*.

Rancangan antarmuka untuk fitur mendaftarkan data tangki dan memperbarui data tangki hamper sama. Antarmuka tersebut berupa *form* yang harus diisi oleh pengguna.

Rancangan antarmuka untuk fitur melihat riwayat monitoring tangki hamper sama dengan monitoring tangki, namun pada fitur ini, data tangki akan ditampilkan mulai dari sesaat setelah kendaraan berangkat hingga kendaraan tersebut sampai di tujuan. Rancangan antar muka untuk fitur ini dapat dilihat pada gambar 3.10.



***Gambar 3.10. Rancangan Antarmuka untuk fitur lihat riwayat monitoring tangki***

## **BAB IV IMPLEMENTASI**

Pada bab ini akan dibahas mengenai implementasi yang dilakukan berdasarkan rancangan yang telah dijabarkan pada bab sebelumnya. Sebelum penjelasan implementasi akan ditunjukkan terlebih dahulu lingkungan untuk melakukan implementasi.

### **4.1 Lingkungan Implementasi**

Pada implementasi sistem, digunakan beberapa perangkat pendukung sebagai berikut:

1. Arduino UNO R3 dan GSM/GPRS/GPS Shield V3.0
2. Sensor Ultrasonik SRF 02 dan *Bluetooth* HC 05 serta HC 06
3. Laptop ASUS intel Pentium dual-core *processor* (2.10 Ghz) dengan 4 GB RAM
4. *Arduino Development Kit* versi 1.6.3 sebagai IDE untuk mengimplementasikan kode program arduino
5. Sublime sebagai *text editor* untuk mengimplementasikan kode program aplikasi monitoring
6. 3 buah baterai 9 V
7. 1 buah *SIM Card*

### **4.2 Implementasi**

Pada subbab ini akan dijelaskan implementasi yang telah dilakukan pada sistem. Terdapat dua lingkup implementasi yang akan dijelaskan pada subbab ini, yaitu implementasi mikrokontroler dan komponen sensor, dan implementasi aplikasi monitoring. Pada masing-masing lingkup tersebut juga akan dijelaskan mengenai

fungsi-fungsi utama yang digunakan sehingga fungsionalitas sistem dapat bekerja dengan baik.

#### 4.2.1 Implementasi Mikrokontroler dan Komponen Sensor

Pada mikrokontroler arduino, secara *default* program harus memiliki fungsi *setup* dan *loop*. Fungsi *setup* berguna untuk pengaturan mikrokontroler pada saat mikrokontroler tersebut melakukan *booting*, sedangkan fungsi *loop* berisi tentang perintah-perintah yang akan dieksekusi ketika mikrokontroler tersebut sedang *running*.

Mengacu pada bab 3 tentang perancangan mikrontroler dan komponen sensor, komponen perangkat keras akan dibagi menjadi dua, yaitu *child node* dan *center node*. Masing-masing komponen memiliki fungsi tersendiri sesuai dengan kegunaan komponen tersebut.

##### 4.2.1.1 Implementasi *Child Node*

Pada *child node*, fungsi yang diimplementasikan adalah fungsi untuk menangkap data dari sensor ultrasonik dan mengirimkannya ke *center node*. Pada fungsi *setup*, node ini akan melakukan koneksi ke *center node* melalui modul *Bluetooth* berdasarkan *MAC Address* dari *Bluetooth* yang ada pada *center node*. *Pseudocode* Fungsi *setup* pada *child node* dapat dilihat pada Gambar 4.1.

1	check AT command connection
2	reset module
3	remove paired device
4	initial module start
5	set link device ← "98D3,31,2018A1"

**Gambar 4.1. Fungsi setup pada child node**

Pada fungsi *loop*, node ini akan men-*trigger* sensor ultrasonik untuk memancarkan sinyal ultrasonik dan menangkap data ketinggian (dalam *centimeter*) lalu mengirimkannya pada *center node*. *Pseudocode* Fungsi *loop* pada *child node* dapat dilihat pada Gambar 4.2.

1	begin transmit signal
2	delay $\leftarrow$ 0.07 second
3	begin read signal
4	<b>if</b> signal available <b>then</b>
5	reading $\leftarrow$ convert signal to byte
6	message $\leftarrow$ header + reading + footer
7	bluetooth_send(message)
8	delay $\leftarrow$ 10 second

**Gambar 4.2. Implementasi fungsi loop pada child node**

#### 4.2.1.2 Implementasi Center Node

Pada *center node* terdapat beberapa fungsi tambahan selain fungsi *setup* dan fungsi *loop*, karena selain menerima data dari *child node*, *center node* juga bertugas untuk tersambung dengan satelit sehingga dapat menerima data GPS, serta bertugas untuk mengirimkan data ke server. Fungsi tambahan untuk *center node* antara lain fungsi *read\_GPS*, *convert2degrees*, *send\_data*, dan *getHeight*.

Fungsi *read\_GPS* bertugas untuk menerima data GPS dari satelit, dengan mengeksekusi *AT Command* yaitu *AT+CGPSINF=0*. Setelah mengeksekusi perintah tersebut, maka mikrokontroler akan menerima data mentah dari satelit yang harus diolah sehingga menghasilkan data koordinat yang valid. Fungsi yang bertugas mengolah data mentah tersebut adalah fungsi *convert2degrees*. *Pseudocode* fungsi *read\_GPS* dan *convert2degrees* dapat dilihat pada Gambar 4.3 dan Gambar 4.4.

1	<code>answer ← send_at_command("AT+CGPSINF=0")</code>
2	<code>if answer == 1 then</code>
3	<code>input_data ← read_string()</code>
4	<code>strtok(input_data, ",")</code>
5	<code>longitude ← strtok(NULL, ",")</code>
6	<code>latitude ← strtok(NULL, ",")</code>
7	<code>convert2degrees(latitude)</code>
8	<code>convert2degrees(longitude)</code>
9	<code>else</code>
	<code>error retrieve GPS data</code>

*Gambar 4.3. Fungsi read\_GPS*

1	<code>retrieve input</code>
2	
3	<code>if input[0] == '-' then</code>
4	<code>negative ← true</code>
5	<code>aux ← strtok(input+1, ".")</code>
6	<code>else</code>
7	<code>aux ← strtok(input+1, ".")</code>
8	
9	<code>degree ← atof(aux)</code>
10	<code>aux ← strtok(NULL, "\0")</code>
11	<code>minutes ← atof(aux)</code>
12	<code>minutes ← minutes / 1000000</code>
13	
14	<code>if degree &lt; 100 then</code>
15	<code>minutes ← minutes + degree</code>
16	<code>else</code>
17	<code>minutes ← int(degree) % 100</code>
18	<code>degree ← int(degree) / 100</code>
19	
20	<code>degree ← degree + minutes / 60</code>
21	
22	<code>if negative == true then</code>
23	<code>degree ← degree * -1.0</code>
24	
25	<code>number_float ← degree</code>
26	<code>new_number ← (long)number_float</code>
27	<code>size ← 0</code>

```

28
29  while true then
30      size ← size + 1
31      digit ← new_number
32      new_number ← new_number / 10
33      int_part[size-1] ← digit
34      if new_number == 0 then
35          break
36
37  index ← 0
38  if negative is true then
39      index ← index + 1
40      input[0] ← "-"
41  for i ← size-1 to 0
42      input[index] ← int_part[i] + '0'
43      index ← index+1
44
45  input[index] ← '.'
46  index ← index +1
47  number_float ← number_float -
    (int)number_float
48  for i ← 1 to 6
49      number_float ← number_float * 10
50      digit ← (long)number_float
51      number_float ← number_float - digit
52      input[index] ← char(digit) + 48
53      index ← index+1
54  input[index] ← '\0'
55
56  return input

```

**Gambar 4.4. Fungsi convert2degrees**

Fungsi *getHeight* bertugas untuk mendapatkan data dari *string* yang diterima dari *child node*. Sedangkan fungsi *send\_data* bertugas untuk mengirimkan data ketinggian, *latitude*, *longitude*, dan id kendaraan ke server. *Pseudocode* fungsi *getHeight* dan *send\_data* dapat dilihat pada Gambar 4.5 dan Gambar 4.6.

1	<code>index_start ← message.indexOf(node_source)</code>
---	---



	+ 6
2	index_end ← message.indexOf(end_data)
3	message ← message.substring(index_start,
4	index_end).toFloat()
5	
6	<b>return</b> message

**Gambar 4.5. Fungsi getHeight**

1	answer ← send_at_command("AT+HTTPIPINIT")
	<b>if</b> answer == 1 <b>then</b>
2	answer ←
	send_at_command("AT+HTTTPARA=CID,1")
3	<b>if</b> answer == 1 <b>then</b>
4	param_str ← bundle_data
5	<b>print</b> param_str
6	delay ← 2 second
7	answer ←
	send_at_command("AT+HTTTPACTION=0")
8	<b>if</b> answer == 1 <b>then</b>
9	<b>print</b> "done"
10	<b>else</b>
11	<b>print</b> "error send data"
12	<b>else</b>
13	<b>print</b> "error setting CID"
14	<b>else</b>
15	<b>print</b> "error initializing"
16	send_at_command("AT+HTTPTERM")

**Gambar 4.6. Fungsi send\_data**

Pada *center node*, fungsi *setup* berisi perintah untuk *register sim card* sehingga *sim card* tersebut dapat digunakan untuk akses internet. Selain itu juga berisi perintah untuk menyalakan GPS dan menunggu hingga GPS sudah tersambung dengan satelit. *Pseudocode* fungsi *setup* pada *center node* dapat dilihat pada Gambar 4.7.

1	enable GSM mode
2	delay $\leftarrow$ 20 second
3	register_gsm()
4	<b>while</b> start_gps() == 0 <b>then</b>
5	<b>continue</b>
6	
7	<b>while</b> Serial.available() > 0 <b>then</b>
8	Serial.read()
9	
10	<b>print</b> "listening data from node1 and node2.."

***Gambar 4.7. Fungsi setup pada center node***

Fungsi *loop* pada *center node* merupakan fungsi inti dari integrasi komponen sensor yang dirancang. Karena pada fungsi inilah semua data terintegrasi dan dikirim ke server. *Pseudocode* fungsi *loop* pada *center node* dapat dilihat pada Gambar 4.8.

1	<b>Serial.flush()</b>
2	<b>while</b> incoming_data.available() <b>then</b>
	message $\leftarrow$ <b>char</b> (incoming_data.read())
3	
4	<b>if</b> node1 data retrieved <b>and</b> node2 data retrieved
5	data1 $\leftarrow$ getHeight(message, header, footer)
6	data2 $\leftarrow$ getHeight(message, header, footer)
7	average_height $\leftarrow$ (data1 + data2) / 2
8	message $\leftarrow$ " "
9	delay $\leftarrow$ 2 second
10	read_gps()
11	send_data()
12	delay $\leftarrow$ 3 second

***Gambar 4.8. Fungsi loop pada center node***

## 4.2.2 Implementasi Aplikasi Monitoring

Seperti yang sudah dijelaskan pada bab 3, terdapat 4 fitur utama pada aplikasi monitoring, yaitu monitoring tangki, mendaftarkan data tangki, memperbarui data tangki, dan melihat riwayat data tangki. Pada subbab ini akan dijelaskan implementasi dari masing-masing fitur.

### 4.2.2.1 Implementasi Fitur Monitoring Tangki

Pada fitur ini, terdapat fungsi utama yang dijalankan, yaitu fungsi *getVehicleData*. Fungsi tersebut akan mengambil data pada database, mengolah data tersebut, dan mengirimkan data dalam bentuk *JSON* ke *view* sehingga dapat dibaca oleh pengguna. Fungsi tersebut juga akan memanggil fungsi *calculateVolume* yang bertugas untuk menghitung volume tangki berdasarkan data ketinggian yang diterima dari komponen sensor, dan berdasarkan tipe tangki dari kendaraan yang dimonitoring. *Pseudocode* fungsi *getVehicleStatus* dan *calculateVolume* dapat dilihat pada Gambar 4.9 dan Gambar 4.10.

1	<b>load model</b>
2	query ← get_vehicle_data
3	data ← calculate_volume
4	<b>return json_encode</b> (data)

**Gambar 4.9. Fungsi *getVehicleStatus***

1	<b>load model</b>
2	<b>foreach</b> row ← query
3	<b>if</b> row["tank_type"] == "Tube" <b>then</b>
4	current_volume ← 3,14 * (row["diameter"] / 2) * (row["liquid_level"] / 2) * row["longsize"]
5	<b>else if</b> row["tank_ype"] == "Rectamgular" <b>then</b>

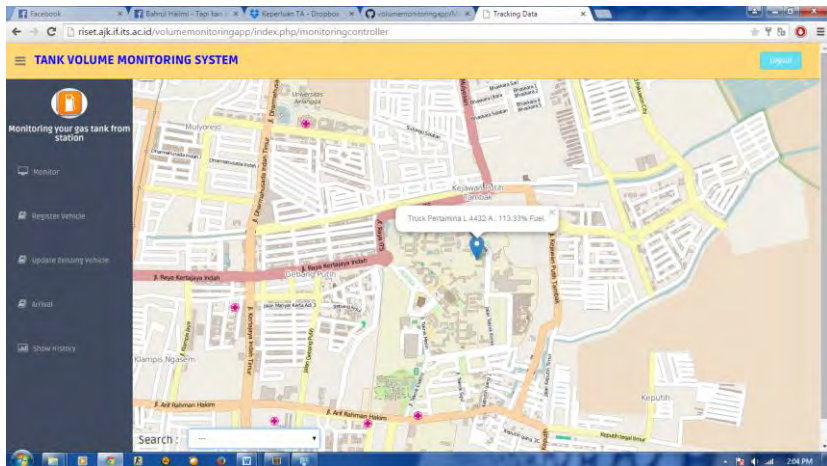
6	<code>current_volume ← \$row["liquid_level"] * row["longsize"] * row["diameter"]</code>
7	<code>initial_volume ← get_initial_data(vehicle_id)</code>
8	<code>percentage ← current_volume / initial_volume * 100</code>
9	<code><b>array_push</b>(query, current_volume, initial_volume, percentage)</code>
10	<code><b>return</b> query</code>

**Gambar 4.10. Fungsi calculateVolume**

Agar fungsi tersebut dapat dijalankan secara periodik oleh *view* tanpa memuat ulang halaman tersebut, maka akan digunakan *jquery* untuk menjalankan fungsi tersebut. Lalu data *json* yang diterima dari fungsi *getVehicleStatus* akan diolah dan digambar di peta pada halaman monitoring tangki. Proses tersebut akan dilakukan secara periodic dengan interval 5 detik. *pseudocode*Proses penggambaran pada peta menggunakan *jquery* dapat dilihat pada Gambar 4.11. Halaman fitur monitoring tangki dapat dilihat pada Gambar 4.12.

1	<code>remove available markers</code>
2	<code>data ← get_json</code>
3	<code><b>foreach</b> key ← data</code>
4	<code>status ← key.vehicle_name + key.current_volume + key.initial_volume</code>
5	<code><b>if</b> key.latitude == 0 <b>or</b> key.longitude == 0 <b>then</b></code>
6	<code>status ← status + "unknown location"</code>
7	<code>marker ← set_default_position</code>
8	<code><b>else</b></code>
9	<code>marker ← set_location(key.latitude, key.longitude)</code>
10	<code>draw(marker, status)</code>

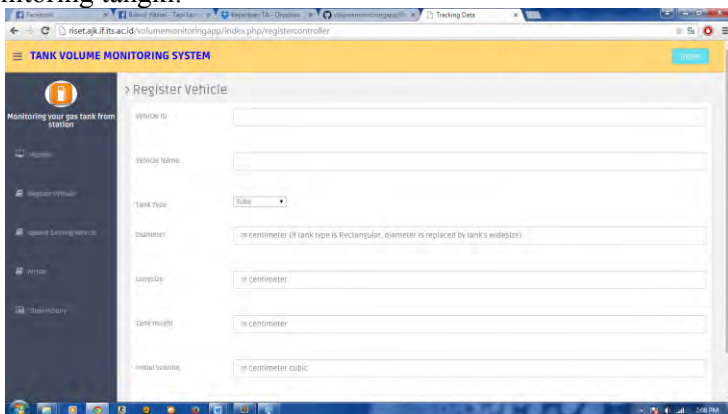
**Gambar 4.11. Proses penggambaran icon pada peta dengan menggunakan jquery**



**Gambar 4.12. Halaman fitur monitoring tangki**

#### 4.2.2.2 Implementasi Fitur Mendaftarkan Data Tangki dan Memperbarui Data Tangki

Pada fitur ini, terdapat fungsi untuk memasukan data dari form yang *diinput* oleh pengguna dan memasukkannya kedalam database. Data tangki tersebut akan digunakan untuk perhitungan data pada monitoring tangki.



**Gambar 4.13. Halaman fitur mendaftar data tangki**

Perbedaan antara mendaftar baru dan memperbarui kendaraan adalah, jika memperbarui kendaraan, maka pengguna hanya perlu memilih id kendaraan yang telah terdaftar dan hanya perlu mengganti data yang ingin diganti saja. Halaman fitur mendaftarkan data tangki dapat dilihat pada Gambar 4.13 dan halaman fitur mendaftarkan data tangki dapat dilihat pada Gambar 4.14.

The screenshot shows a web application titled 'TANK VOLUME MONITORING SYSTEM'. The main content area is titled '> Update Vehicle'. It contains a form with the following fields:

- Select ID: A dropdown menu.
- Vehicle Name: A text input field.
- Tank Type: A dropdown menu with 'Tube' selected.
- Diameter: A text input field with placeholder text: 'In centimeter (If tank type is Rectangular, diameter is replaced by tank's width)
- Length: A text input field with placeholder text: 'In centimeter
- Tank Weight: A text input field with placeholder text: 'In centimeter
- Initial Volume: A text input field with placeholder text: 'In centimeter Cubic

A 'Submit' button is located at the bottom left of the form. On the left side, there is a sidebar with a logo and the text 'Monitoring your gas tank from station'. The sidebar contains the following menu items: Monitor, Add New Vehicle, Update Vehicle, Logout, and Inventory.

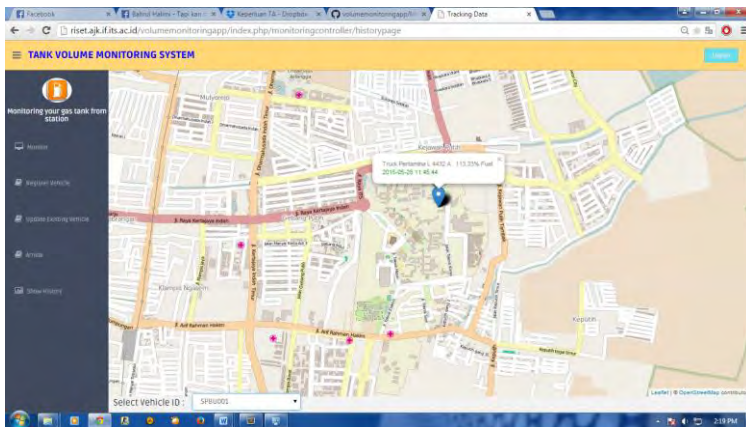
**Gambar 4.14. Halaman fitur memperbarui data tangki**

#### 4.2.2.3 Implementasi Fitur Melihat Riwayat Tangki

Fitur ini memiliki fungsi utama yaitu *showHistory* yang juga memanggil fungsi *calculateVolume* untuk perhitungan volume pada tangki. Perbedaan fitur ini dengan fitur monitoring tangki adalah, pada fitur ini, semua *track record* dari tangki mulai dari berangkat hingga sampai tujuan akan ditampilkan, dan yang bisa ditampilkan dalam halaman ini adalah kendaraan yang tidak sedang dimonitoring. *Pseudocode* fungsi *showHistory* dapat dilihat pada Gambar 4.15.

1	<code>vid ← input_get(vehicle_id)</code>
2	<code>query ← get_history(vid)</code>
3	<code>data ← calculateVolume(query)</code>
4	<code>return json_encode(data)</code>

**Gambar 4.15. Fungsi showHistory**



**Gambar 4.16. Halaman melihat riwayat kendaraan**

#### 4.2.2.4 Implementasi Web Service

*Web Service* yang diimplementasikan pada aplikasi ini merupakan sebuah fungsi yang dapat dipanggil menggunakan *method GET* pada *HTTP*. Fungsi inilah yang dipanggil oleh mikrokontroler untuk mengirim data ke server. *Pseudocode* implementasi *web service* dapat dilihat pada Gambar 4.17.

1	<code>load model</code>
2	<code>vehicle_id ← input_get("vehicle_id")</code>
3	<code>latitude ← input_get("latitude")</code>
4	<code>longitude ← input_get("longitude")</code>
5	<code>timestamp ← date_now()</code>
6	<code>height ← initial tank height -</code>

7	<pre>input_get(height) model-&gt;insert_data()</pre>
---	--

***Gambar 4.17. Implementasi web service***



## **BAB V**

### **UJI COBA DAN EVALUASI**

Pada bab ini akan dijelaskan uji coba yang dilakukan pada aplikasi yang telah dikerjakan serta analisa dari uji coba yang telah dilakukan. Pembahasan pengujian meliputi lingkungan uji coba, skenario uji coba yang meliputi uji tingkat akurasi dan uji kinerja serta analisa setiap pengujian.

#### **5.1 Lingkungan Uji Coba**

Pada subbab ini menjelaskan tentang lingkungan yang digunakan untuk menguji fungsionalitas dan tingkat akurasi dari implementasi sistem yang telah dirancang. Lingkungan uji coba menggunakan bantuan perangkat lunak dan perangkat keras dengan spesifikasi sebagai berikut:

1. Perangkat Keras
  - a. Laptop dengan prosessor Intel Core i3 CPU @
  - b. RAM : 4,00 GB
  - c. Tipe sistem : 64-bit sistem operasi
2. Perangkat Lunak
  - a. Sistem Operasi : *Windows 8.1*
  - b. *IDE* : *arduino 1.6.3*
  - c. Sebuah *web browser*

Spesifikasi prototipe tangki yang digunakan untuk uji coba komponen sensor adalah sebagai berikut:

1. Tinggi : 48.5 cm
2. Panjang : 65.4 cm
3. Lebar : 32.7
4. Ketinggian air : 15 cm
5. Volume air : 32078.7 cm<sup>3</sup>



**Gambar 5.1. Prototipe tangki uji coba**

Untuk sumber daya pada masing-masing node (*center node & child node*), menggunakan 3 buah baterai 9V.

## **5.2 Skenario Uji Coba**

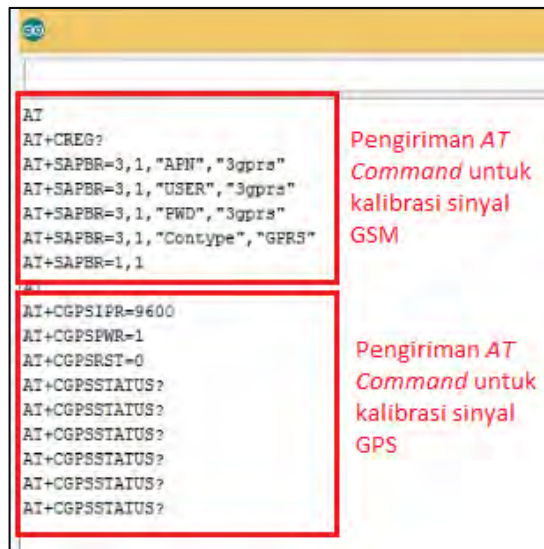
Uji coba ini dilakukan untuk menguji apakah fungsionalitas program telah diimplementasikan dengan benar dan berjalan sesuai rencana. Pada sistem ini, akan dilakukan uji coba fungsionalitas sistem dan uji coba performa atau tingkat akurasi sistem.

### **5.2.1 Uji Coba Fungsionalitas**

Pada uji coba fungsionalitas akan diuji fungsi-fungsi utama pada komponen sensor dan aplikasi monitoring. Keberhasilan pada uji coba ini akan menunjang kualitas dari uji coba yang selanjutnya yaitu uji coba performa atau tingkat akurasi sistem.

### 5.2.1.1 Uji Coba Kalibrasi Sinyal GSM dan GPS

Agar sistem dapat bekerja dengan baik, maka komponen sensor khususnya *center node* harus tersambung dengan jaringan GSM. Selain itu, *center node* juga perlu terhubung dengan satelit agar mendapatkan koordinat lokasi yang valid. Maka dari itu diperlukan kalibrasi agar komponen sensor dapat melakukan dua hal tersebut. Kartu GSM yang digunakan untuk *center node* adalah kartu dari provider *three*. Agar dapat berjalan dengan maksimal, uji coba kalibrasi sinyal GSM dan GPS dilakukan diluar ruangan. *Center node* akan melakukan kalibrasi sinyal pada saat komponen tersebut melakukan *booting*, sehingga berhasil atau tidaknya uji coba ini ditentukan saat komponen tersebut berhasil melakukan *booting* atau tidak. Kalibrasi sinyal GSM dan GPS pada saat *center node* melakukan *booting* dapat dilihat pada Gambar 5.2.



**Gambar 5.2. Kalibrasi sinyal pada saat booting**

Perintah *AT+CREG?* digunakan untuk mengecek status jaringan. Perintah selanjutnya yaitu *AT+SAPBR* digunakan untuk mendaftarkan kartu GSM yang digunakan pada *center node*, agar kartu tersebut dapat digunakan untuk mengirim data ke server melalui jaringan internet. Selanjutnya, perintah *AT+CGPSIPR* digunakan untuk mengatur *baud rate* dari GPS tersebut, lalu untuk menyalakan modul GPS digunakan perintah *CGPSPWR=1*. Perintah *CGPSRST=0* digunakan untuk me-reset modul GPS. *AT+CGPSSTATUS?* dilakukan untuk menghubungkan node tersebut dengan satelit, perintah ini dijalankan berulang kali sampai node tersebut terhubung dengan satelit.

Waktu yang diperlukan untuk mendapatkan sinyal GSM sangatlah cepat, namun untuk terhubung dengan satelit agar dapat menerima data GPS diperlukan waktu yang relatif lebih lama, yaitu berkisar antara 1 – 3 menit, tergantung posisi *center node* diletakkan. Waktu tercepat untuk *center node* dapat terhubung dengan satelit adalah ketika *center node* diletakkan ditempat yang terbuka, atau tidak ada pembatas antara *center node* dengan langit.

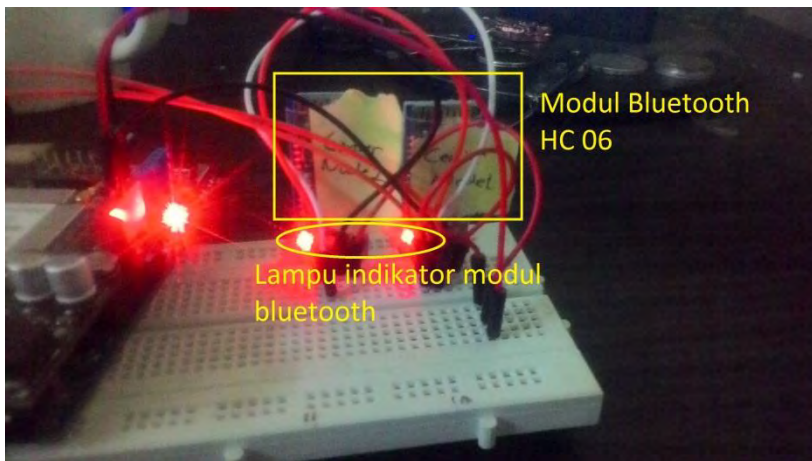
```
AT+CGPSSTATUS?
AT+CGPSSTATUS?
AT+CGPSSTATUS?
AT+CGPSSTATUS?
Listening data from node1 and node2..
```

**Gambar 5.3. Center node berhasil melakukan booting**

### 5.2.1.2 Uji Coba Fungsionalitas *Bluetooth* dan Sensor Ultrasonik

Uji coba ini bertujuan untuk mengetahui apakah jaringan nirkabel sederhana yang dirancang sudah berjalan dengan baik. Seperti yang sudah dijelaskan di bab-bab sebelumnya, komponen sensor terdiri dari 3 node yang tiap nodenya akan saling tersambung melalui modul *Bluetooth* yang dipasang pada masing-masing node.

Untuk mengetahui apakah *Bluetooth* tersebut telah tersambung dengan perangkat *Bluetooth* yang lain adalah dengan cara melihat lampu indikator yang ada pada modul tersebut, jika lampu indikator pada *Bluetooth* menyala berkedip-kedip, maka *Bluetooth* tersebut masih belum tersambung, namun jika lampu indikator menyala terus maka menandakan bahwa *Bluetooth* tersebut sedang tersambung dengan perangkat lain.



***Gambar 5.4. Lampu indikator pada Bluetooth***

Untuk mengetahui apakah *Bluetooth* benar-benar sudah bekerja dengan baik, maka perlu dilakukan transfer data antar node. Data yang ditransfer adalah data ketinggian yang didapat dari sensor ultrasonik pada *child node*, sehingga uji coba ini juga sekaligus mengetahui apakah data ketinggian yang didapat oleh sensor ultrasonik sudah benar atau tidak.

Pada awalnya, sensor ultrasonik (dalam hal ini adalah sensor tipe SRF 02) akan digunakan untuk mengukur jarak antara permukaan air dan dasar tangki, namun ternyata karena terhambat oleh media air, sinyal ultrasonik yang dipancarkan oleh sensor tidak dapat memantul sempurna, sehingga data ketinggian yang didapatkan tidak valid. Untuk mengatasi hal ini, maka sensor

ultrasonik akan dibalik sehingga digunakan untuk mengukur jarak antara permukaan air dan atap tangki, dan untuk mendapatkan kedalaman cairan, data ketinggian dari sensor tersebut akan digunakan sebagai nilai pengurang terhadap tinggi tangki.

Pada *child node*, node tersebut akan ter-trigger untuk mentransfer data ke *center node* ketika sensor ultrasonik berhasil mendapatkan data ketinggian. Pada *center node*, data ketinggian yang diterima dari kedua *child node* akan dirata-rata, dan ketika *center node* telah mendapatkan data dari kedua *child node*, maka *center node* akan ter-trigger untuk meminta data GPS ke satelit, membungkus kedua data tersebut, dan mengirimkannya ke server.

```
AT+CGPSINF=0
AT+HTTPINIT
AT+HTTPPARA="CID",1
AT+HTTPPARA="URL","http://riset.ajk.if.its.ac.id/"
AT+HTTPACTION=0
Done
AT+HTTPTERM
AT+CGPSINF=0
```

**Gambar 5.5. Center node berhasil menerima data dari child node**

```
getrecorddata?height=33.500&latitude=-7.279424&longitude=112.797187&vehicle_id=SPBU001"
```

**Gambar 5.6. Bundle data yang dikirim ke server**

Gambar 5.5 merupakan tampilan dari serial monitor pada *center node*, sedangkan Gambar 5.6 merupakan *bundle* data yang dikirim ke server. Dapat dilihat pada gambar 5.5 bahwa *center node* telah ter-trigger untuk menjalankan beberapa *AT+Command*, karena telah berhasil terkoneksi dan menerima data yang dikirim dari *child node*. Pada gambar tersebut terdapat beberapa *AT+Command* untuk

mengirimkan data ke server, diantaranya adalah *AT+HTTPACTION=0* yang berguna untuk mengeksekusi perintah *GET* pada *HTTP*. Data-data yang dikirim ke server akan disimpan pada tabel *track\_record*, namun sebelum disimpan pada tabel tersebut, data ketinggian yang diterima akan dijadikan pengurang terhadap tinggi tangki, sehingga didapatkan data kedalaman cairan yang valid. Gambar tabel *track\_record* yang berhasil terisi data oleh komponen sensor dapat dilihat pada Gambar 5.7.

id	vehicle_id	latitude	longitude	altitude	liquid_level	timestamp
20	SPBU001	-7.279467	112.796997	0	17	2015-05-25 02:58:24
21	SPBU001	-7.279467	112.796997	0	16.5	2015-05-25 02:58:57

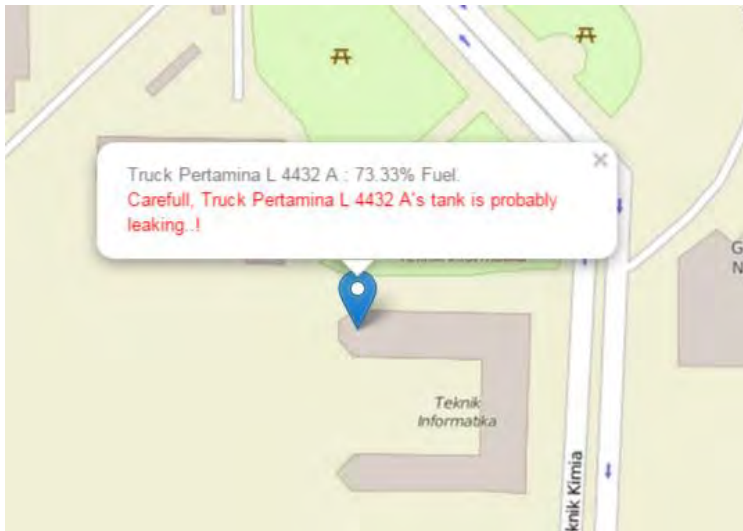
**Gambar 5.7. Data yang berhasil terkirim ke server disimpan pada tabel *track\_record***

### 5.2.1.3 Uji Coba Fitur Monitoring Tangki pada Aplikasi Monitoring

Uji coba ini dilakukan untuk menguji fitur utama pada aplikasi monitoring. Dari fitur ini, dapat dilihat hasil akhir dari kinerja komponen sensor yang dipasang pada prototipe tangki. Seperti yang sudah dijelaskan pada bab 3 dan bab 4, fitur ini akan mengolah dan menampilkan data yang ada pada tabel *track\_record* agar dapat dilihat oleh pengguna.

Sebelum masuk ke halaman monitoring tangki, pengguna diharuskan *login* terlebih dahulu, agar tidak semua orang dapat mengakses fitur ini. Setelah *login*, pengguna langsung diarahkan ke halaman utama monitoring tangki. Di halaman tersebut, seperti yang sudah dijelaskan pada bab 4, terdapat *display* peta dunia seperti halnya tampilan yang ada pada *website* [maps.google.com](http://maps.google.com), namun pada aplikasi ini menggunakan *leaflet.js* sebagai API untuk menampilkan peta dunia tersebut. Program akan menjalankan fungsi *getVehicleStatus* setiap 5 detik sekali untuk mengambil data kendaraan yang ada pada tabel *track\_record* dan menampilkannya pada halaman tersebut pada koordinat yang sesuai dengan data

*latitude* dan *longitude* yang ditangkap oleh komponen sensor. Untuk data volume tangki yang ditampilkan pada halaman tersebut adalah persentase perbandingan antara volume awal sebelum kendaraan berangkat dengan volume saat ini. Gambar 5.8 menunjukan bahwa fitur monitoring tangki dapat berjalan sesuai dengan fungsionalitas yang diinginkan.



**Gambar 5.8. Fitur monitoring tangki berhasil mengolah data yang ada pada tabel *track\_record***

### 5.2.2 Uji Coba Performa

Pada bagian ini akan dilakukan uji coba performa atau tingkat akurasi komponen sensor ketika diimplementasikan pada keadaan *real time*. Uji coba performa akan menggunakan prototipe tangki dengan ukuran yang sudah dideskripsikan pada lingkungan uji coba, dan uji coba akan dilakukan di lingkungan gedung Teknik Informatika ITS. Terdapat 3 skenario uji coba yang dilakukan, yang akan dijelaskan pada masing-masing subbab.



Uji coba performa menggunakan baterai 9V untuk masing-masing node pada komponen sensor. Untuk *child node*, baterai 9V dapat bertahan lama, namun untuk *center node* baterai 9V cepat habis karena pada *center node* memerlukan banyak daya untuk menyalakan modul GPS, modul GPRS dan modul GSM, sehingga baterai hanya dapat bertahan dalam rentang waktu kurang lebih 2 menit..



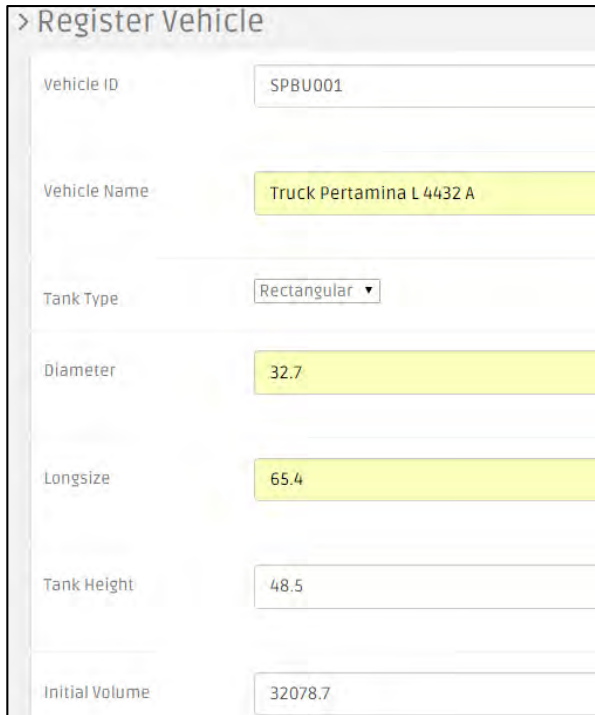
***Gambar 5.9. Child node yang diberi pelampung sehingga bisa mengapung di air***



***Gambar 5.10. Center node***

Sebelum alat sensor dipasang pada prototipe tangki, data prototipe tangki harus didaftarkan dulu pada fitur mendaftarkan data

kendaraan, agar prototipe tangki tersebut dapat dimonitoring. Halaman mendaftarkan data prototipe tangki dapat dilihat pada Gambar 5.11.



The image shows a web form titled "> Register Vehicle". It contains several input fields with labels on the left and values in the input boxes. The values for 'Vehicle Name', 'Diameter', 'Longsize', and 'Initial Volume' are highlighted in yellow. 'Tank Type' is a dropdown menu showing 'Rectangular'.

Field Label	Value
Vehicle ID	SPBU001
Vehicle Name	Truck Pertamina L 4432 A
Tank Type	Rectangular
Diameter	32.7
Longsize	65.4
Tank Height	48.5
Initial Volume	32078.7

***Gambar 5.11. Mendaftarkan data prototipe tangki***

#### **5.2.2.1 Skenario Uji Coba 1**

Skenario uji coba satu adalah uji coba akurasi komponen sensor jika tangki dalam keadaan normal.



***Gambar 5.12. Keadaan tangki untuk uji coba 1***

Skenario uji coba 1 dilakukan selama 70 detik, dan berhasil mengirimkan 5 data ke server. Data ketinggian dan data lokasi yang berhasil dikirim dapat dilihat pada Tabel 5.1 dan Tabel 5.2.

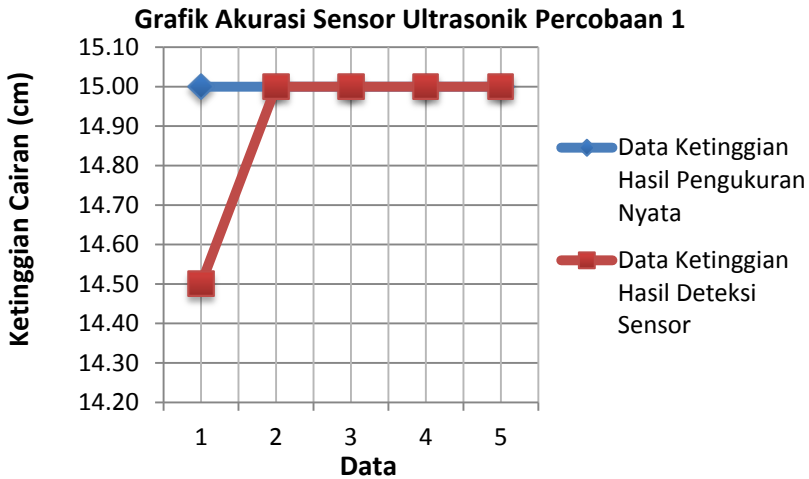
***Tabel 5.1. Data ketinggian yang berhasil dikirim ke server pada uji coba 1***

No	Hasil deteksi komponen sensor (dalam cm)	Setelah kalkulasi (dalam cm)
1	34	14.5
2	33.5	15
3	33.5	15
4	33.5	15
5	33.5	15

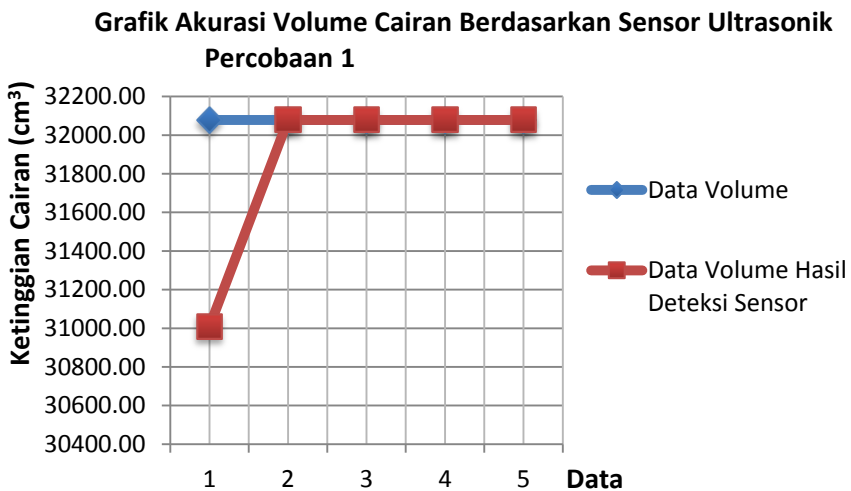
***Tabel 5.2. Data lokasi yang berhasil dikirim ke server pada uji coba 1***

No	Hasil Deteksi Sensor		Koordinat Nyata	
	Latitude	Longitude	Latitude	Longitude
1	-7.279424	112.797187	-	-
2	-7.279424	112.797187	-	-
3	-7.279424	112.797187	-	-
4	-7.279424	112.797187	-	-
5	-7.279424	112.797187	-	-

Setelah melakukan percobaan, maka harus dilakukan evaluasi antara data ketinggian nyata yang ada pada *database* dan data ketinggian yang didapatkan dari hasil sensor ultrasonik. Selain itu juga perlu dilakukan evaluasi antara data volume nyata yang ada pada database dan data volume hasil perhitungan berdasarkan data ketinggian hasil sensor ultrasonik. Grafik hasil akurasi dari sensor ultrasonik pada percobaan 1 dapat dilihat pada gambar 5.12 dan grafik hasil akurasi volume cairan berdasarkan data ketinggian hasil sensor ultrasonik dapat dilihat pada Gambar 5.13.



*Gambar 5.13. Grafik akurasi sensor ultrasonic pada percobaan 1*



*Gambar 5.14. Grafik akurasi volume cairan berdasarkan sensor ultrasonik pada percobaan 1*

### 5.2.2.2 Skenario Uji Coba 2

Skenario uji coba dua adalah uji coba akurasi komponen sensor jika tangki dalam keadaan dimiringkan, atau seolah-olah jika kendaraan sedang berada dalam tanjakan atau turunan. Sudut kemiringan adalah  $10^0$ . Gambar keadaan tangki pada uji coba 2 dapat dilihat pada Gambar 5.15.



**Gambar 5.15. Keadaan tangki untuk skenario uji coba 2**

Skenario uji coba 2 dilakukan selama 75 detik, dan berhasil mengirimkan 5 data ke server. Data ketinggian dan data lokasi yang berhasil dikirim dapat dilihat pada Tabel 5.3 dan Tabel 5.4.

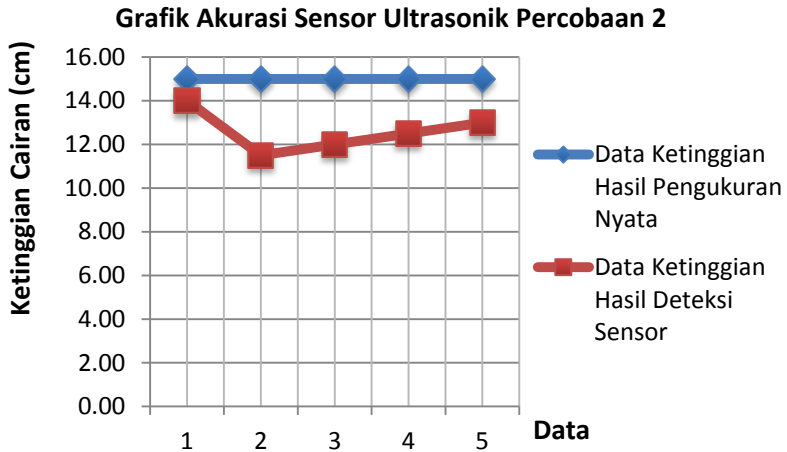
**Tabel 5.3. Data ketinggian yang berhasil dikirim ke server pada uji coba 2**

No	Hasil deteksi komponen sensor (dalam cm)	Setelah kalkulasi (dalam cm)
1	34.5	14
2	37	11.5
3	36.5	12
4	36	12.5
5	35.5	13

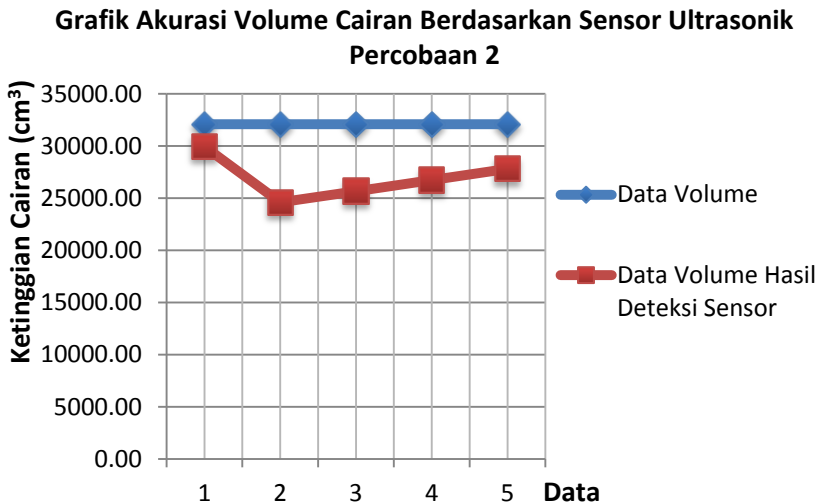
***Tabel 5.4. Data lokasi yang berhasil dikirim ke server pada uji coba 1***

No	Hasil Deteksi Sensor		Koordinyat Nyata	
	Latitude	Longitude	Latitude	Longitude
1	-7.279586	112.79692	-7.279401	112.797137
2	-7.279586	112.79692	-7.279401	112.797137
3	-7.279582	112.796928	-7.279401	112.797137
4	-7.27958	112.797187	-7.279401	112.797137
5	-7.279369	112.796928	-7.279401	112.797137

Setelah melakukan percobaan, maka harus dilakukan evaluasi antara data ketinggian nyata yang ada pada *database* dan data ketinggian yang didapatkan dari hasil sensor ultrasonik. Selain itu juga perlu dilakukan evaluasi antara data volume nyata yang ada pada *database* dan data volume hasil perhitungan berdasarkan data ketinggian hasil sensor ultrasonik. Grafik hasil akurasi dari sensor ultrasonik pada percobaan 2 dapat dilihat pada Gambar 5.15 dan grafik hasil akurasi volume cairan berdasarkan data ketinggian hasil sensor ultrasonik dapat dilihat pada Gambar 5.16.



*Gambar 5.16. Grafik akurasi sensor ultrasonic pada percobaan 2*



*Gambar 5.17. Grafik akurasi volume cairan berdasarkan sensor ultrasonic pada percobaan 2*



### 5.2.2.3 Skenario Uji Coba 3

Skenario uji coba 3 dilakukan untuk mengetahui apakah komponen sensor mampu mendeteksi kebocoran, atau pencurian muatan tangki. Pada uji coba 3 akan diambil 10 data, dan pada tangki akan dilakukan pengurangan muatan secara periodik. Kondisi tangki pada uji coba 3 ini sama seperti pada skenario uji coba 1. Gambar kondisi tangki pada skenario ini dapat dilihat pada Gambar 5.18.



***Gambar 5.18. Prototipe tangki 3 akan dikurangi muatannya secara periodik***

Data ketinggian dan data lokasi yang berhasil dikirim dapat dilihat pada Tabel 5.5.

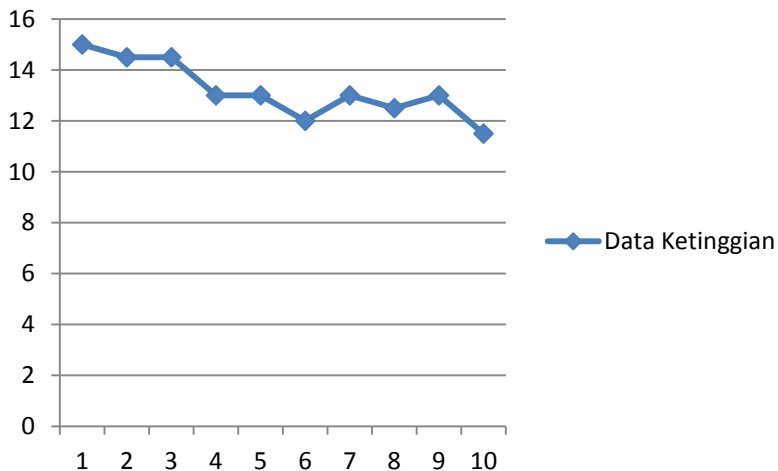
***Tabel 5.5. . Data yang berhasil dikirim ke server pada uji coba 3***

No	Data Ketinggian dari komponen sensor (cm)	Data Ketinggian setelah kalkulasi (cm)	Latitude	Longitude
1	33.5	15	-7.279343	112.797019

2	34	14.5	-7.279335	112.797012
3	34	14.5	-7.279335	112.797012
4	35.5	13	-7.279335	112.797012
5	35.5	13	-7.279335	112.797012
6	36.5	12	-7.279328	112.797012
7	35.5	13	-7.279328	112.797012
8	36	12.5	-7.279335	112.797012
9	35.5	13	-7.279328	112.797012
10	37	11.5	-7.279328	112.797012

Grafik ketinggian cairan pada skenario uji coba 3 dapat dilihat pada Gambar 5.19.

### Grafik Kondisi Tangki Ketika Muatan Bocor / Dicuri



**Gambar 5.19. Grafik kondisi tangki ketika muatan bocor / dicuri**

### 5.3 Analisa Hasil Uji Coba

Berdasarkan skenario uji coba 1 dan 2, medan yang dilalui oleh kendaraan pengangkut tangki cukup berpengaruh pada kemampuan deteksi komponen sensor. Jika kendaraan pengangkut tangki melewati medan tanjakan atau turunan, maka kemampuan deteksi sensor relatif kurang stabil jika dibandingkan dengan ketika kendaraan melewati medan yang landai. Perbandingan tingkat akurasi pada percobaan 1 dan 2 dapat dilihat pada Tabel 5.6.

***Tabel 5.6. Perbandingan tingkat akurasi percobaan 1 dan 2***

	Rata-rata ketinggian (cm)	Rata-rata volume (cm <sup>3</sup> )	Tingkat Akurasi
Uji Coba 1	14.9	31864.84	99.33%
Uji Coba 2	12.6	26946.11	84%

Pada skenario uji coba 3, dapat dilihat bahwa ketika tangki sedang mengalami kebocoran atau muatannya dicuri, maka pola grafik akan mengalami penurunan secara periodik. Sehingga untuk membedakan apakah tangki sedang mengalami kebocoran atau apakah tangki sedang berada di medan yang tidak biasa, maka dapat dilihat dari pola grafik yang ada, jika pola grafik cenderung menurun hingga melewati 50% maka tangki sedang mengalami kebocoran atau sedang dicuri muatannya.

Untuk kasus pencurian dengan mengganti muatan BBM dengan air biasa agar volume cairan di dalam tangki tidak berkurang, maka cara mengatasinya adalah dengan memanfaatkan layar monitoring yang ada pada aplikasi. Kasus pencurian ini dilakukan dengan membawa wadah air tambahan untuk mengganti muatan BBM tersebut, sehingga pencuri memiliki pilihan antara membawa wadah air sebesar minimal setengah dari ukuran tangki yang notabene

sangat besar dan akan ketahuan jika pencuri tersebut melakukan kecurangan, atau pencuri membawa wadah kecil yang telah terisi air dan sudah di sebar di beberapa titik, sehingga pencuri perlu berhenti secara periodik untuk mengambil wadah-wadah tersebut. Untuk mengatasi pilihan kedua tersebut, petugas pemantau hanya perlu melihat pergerakan kendaraan dari layar monitoring. Jika pergerakan kendaraan tidak normal, yaitu sering berhenti secara periodik di titik-titik tertentu, maka kendaraan tersebut perlu dicurigai dan petugas perlu menghubungi kru dari kendaraan tersebut, sehingga kasus pencurian dengan metode ini dapat diatasi.

## **BAB VI**

### **KESIMPULAN DAN SARAN**

#### **6.1 Kesimpulan**

Dari hasil pengamatan dan percobaan selama perancangan, implementasi, dan uji coba aplikasi, maka dapat diambil kesimpulan sebagai berikut:

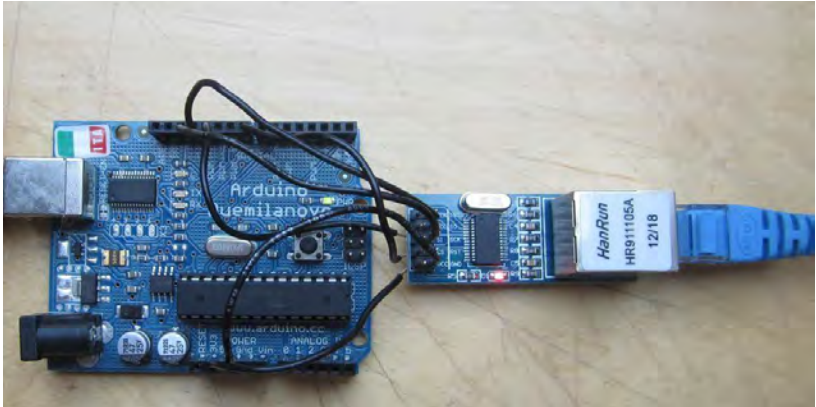
1. Perancangan jaringan nirkabel sederhana dengan memanfaatkan mikrokontroler arduino, modul GSM/GPRS/GPS Shield dan sensor ultrasonik mampu memonitoring keadaan tangki dengan tingkat akurasi terbaik yaitu 99.33 %.
2. Medan yang dilalui kendaraan tangki cukup berpengaruh terhadap kemampuan deteksi komponen sensor, dengan beda tingkat akurasi 99.33% di medan normal dan 84 % di medan tidak normal.
3. Sistem mampu mendeteksi kebocoran atau pencurian muatan tangki sehingga dapat dilakukan pencegahan secara dini terhadap muatan tangki.

#### **6.2 Saran**

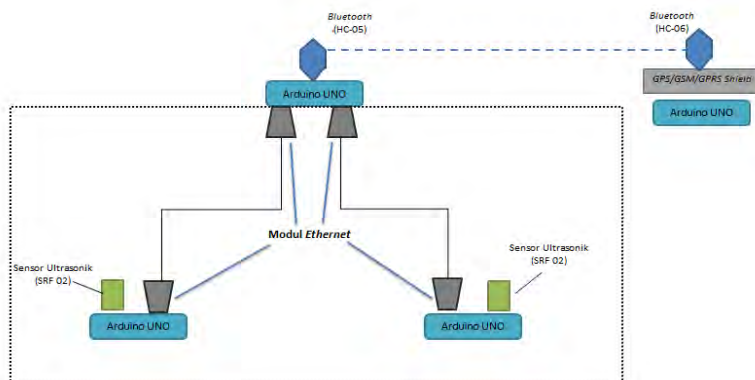
Saran yang diberikan untuk pengembangan aplikasi ini adalah:

1. Untuk meningkatkan nilai akurasi di medan yang tidak normal, maka diperlukan mekanisme supaya alat sensor yang diletakan di air menjadi stabil, seperti contoh mengaitkan pelampung pada dasar tangki dengan menggunakan tali, sehingga pelampung tidak terbawa arus.
2. Untuk meningkatkan nilai akurasi sensor, dapat dicoba dengan menggunakan sensor sonar yang khusus untuk mendeteksi kedalaman air.

3. Sinyal *bluetooth* menjadi tidak stabil ketika terhalangi oleh benda padat seperti dinding tangki, sehingga untuk mengatasi hal ini, dapat memanfaatkan modul *ethernet* (dapat dilihat pada Gambar 6.1) sebagai media komunikasi antar node, sehingga perlu merubah arsitektur jaringan nirkabel sederhana menjadi seperti Gambar 6.2.



**Gambar 6.1. Modul Ethernet**



**Gambar 6.2. Arsitektur sistem jika menggunakan modul ethernet**

## DAFTAR PUSTAKA

- [1] DFRobot, "DFRobot," [Online]. Available: <http://www.dfrobot.com/image/data/TEL0051/3.0/NMEA%20Reference%20Manual-Rev2.1-Dec07.pdf>. [Accessed December 2014].
- [2] B. A. Minartiningtyas, "Informatika - Artikel Teknik Informatika dan Sistem Informasi," 2 Januari 2013. [Online]. Available: <http://informatika.web.id/principal-component-analysis-pcaproyeksi-eigen.htm>. [Accessed 5 Januari 2015].
- [3] "Kelas Robot," Kelas Robot, [Online]. Available: <http://www.kelasrobot.com/2014/12/jenis-jenis-microcontroller-arduino.html>. [Accessed 9 June 2015].
- [4] S. L. C. Instructables Team, "instructables.com," Instructables, [Online]. Available: <http://www.instructables.com/id/AT-command-mode-of-HC-05-Bluetooth-module/>. [Accessed March 2015].
- [5] Libellium, "Cooking Hacks," Libellium, [Online]. Available: <https://www.cooking-hacks.com/projects/arduino-realtime-gps-gprs-vehicle-tracking>. [Accessed March 2015].
- [6] S. S. w. s. Ltd., SIM 908 AT Command Manual\_V1.01, Shanghai: Shanghai SIM Com, 2011.
- [7] K. Mikrokontrol, "Kelas Mikrokontrol," [Online]. Available: <http://www.kelas-mikrokontrol.com/e-learning/mikrokontroler/pengantar-arduino.html>. [Accessed December 2014].
- [8] DFRobot, "DFRobot," [Online]. Available: [http://www.dfrobot.com/wiki/index.php/GPS/GPRS/GSM\\_Module\\_V3.0\\_\(SKU:TEL0051\)](http://www.dfrobot.com/wiki/index.php/GPS/GPRS/GSM_Module_V3.0_(SKU:TEL0051)). [Accessed December 2014].
- [9] K. Elektronika, "Komponen Elektronika," [Online]. Available: <http://komponenelektronika.biz/sensor-ultrasonik.html>. [Accessed December 2014].

- [10] S. Djanali, M. Husni and H. Titi Ciptaningtyas, "Aplikasi Pemantauan Pendistribusian Bahan Bakar Minyak Menggunakan GPS pada Perangkat Android dan Sensor Kecepatan Air Berbasis Mikrokontroler Arduino," in *Seminar Nasional Teknologi Terapan*, 2014.



## LAMPIRAN

*Tabel A.1. Hasil Analisa Uji Coba Skenario 1*

### Hasil Analisa Uji Coba Skenario 1

No	Data Ketinggian Hasil Pengukuran Nyata	Data Ketinggian Hasil Deteksi Sensor	Akurasi (dalam Persen)	Data Volume	Data Volume Hasil Deteksi Sensor	Akurasi
1	15.00	14.50	96.67	32078.70	31009.41	96.67
2	15.00	15.00	100.00	32078.70	32078.70	100.00
3	15.00	15.00	100.00	32078.70	32078.70	100.00
4	15.00	15.00	100.00	32078.70	32078.70	100.00
5	15.00	15.00	100.00	32078.70	32078.70	100.00
<b>Rata-rata</b>	<b>15.00</b>	<b>14.90</b>	<b>99.33</b>	<b>32078.7</b>	<b>31864.842</b>	<b>99.33</b>

*Tabel A.2. Hasil Analisa Uji Coba Skenario 2***Hasil Analisa Uji Coba Skenario 2**

No	Data Ketinggian Hasil Pengukuran Nyata	Data Ketinggian Hasil Deteksi Sensor	Akurasi (dalam Persen)	Data Volume	Data Volume Hasil Deteksi Sensor	Akurasi
1	15.00	14	93.33	32078.70	29940.12	93.33
2	15.00	11.5	76.67	32078.70	24593.67	76.67
3	15.00	12	80.00	32078.70	25662.96	80.00
4	15.00	12.5	83.33	32078.70	26732.25	83.33
5	15.00	13	86.67	32078.70	27801.54	86.67
<b>Rata-rata</b>	<b>15.00</b>	<b>12.60</b>	<b>84.00</b>	<b>32078.7</b>	<b>26946.108</b>	<b>84.00</b>

## BIODATA PENULIS



Yoga Pratama Aliarham, lahir di Malang pada 22 November 1993. Penulis menempuh pendidikan mulai dari TK ABA 26 Malang (1997 - 1999). SD Sawojajar 3 Malang (1999 - 2005), SMPN 21 Malang (2005 - 2008), SMAN 1 Malang (2008 - 2011) dan S1 Teknik Informatika ITS (2011 - 2015).

Selama masa kuliah, penulis aktif dalam organisasi Himpunan Mahasiswa Teknik Computer (HMTTC). Diantaranya adalah menjadi staff departemen pengembangan profesi HMTTC 2012 – 2013 dan kepala departemen pengembangan profesi HMTTC 2013 – 2014. Penulis juga aktif dalam organisasi BEM Fakultas Teknologi Informasi, diantaranya adalah menjadi Pemandu LKMM generasi ke 5. Penulis juga pernah menjadi pembicara dalam beberapa acara di ITS. Diantaranya adalah, menjadi pembicara pada Opentalk Kerja Praktek HMTTC, dan menjadi mentor pengajar pada acara ASEAN Young Enginer and Scientis Summit 2015.

Selama kuliah di teknik informatika ITS, penulis mengambil rumpun mata kuliah Komputasi Berbasis Jaringan (KBJ). Komunikasi dengan penulis dapat melalui email : **aliarham.skom@gmail.com**.